računari

Specijalno izdanje časopisa „Galaksija“
izdaje BIGZ – OOUR „Duga“

11

cena
250
dinara
izlazi
jedanput
mesečno
januar 1986.

- hard diskovi
- video digitajzer
- hakerski marketing
- adresar profi klubova



operativni sistemi „amstradov“ kernal

11

računari

Specijalno izdanje časopisa „Galaksija"

Cena 250 dinara / januar 1986. Izlazi jednom mesečno Izdaje BIGZ OOUR „Duga"

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga"
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-161 (redakcija)
650-528 (prodaja)
651-798 (propaganda)

Generalni direktor
Dobrosav Petrović

Direktor OOUR „Duga"
Bratoljub Babić
Glavni i odgovorni urednik
Gavrilo Vučković
Urednik izdanja
Jova Regasek

Tehnički urednik
Mirko Popov

Redakcija časopisa „Galaksija"
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika
Esad Jakupović, zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Srđan Stojančev, novinar
Gavrilo Vučković, glavni i odgovorni urednik

Stručna saradnja
Dejan Ristanović
Dušan Slavić
Nevenka Spalević
Anđelko Zgorelec
Mihajlo Karapandžić

Autori tekstova:

Nada Aleksić
Branko Đaković
Donat Greber
Vladimir Kostić
Vladimir Krstonošić
Dejan Muhamedagić
Blažimir Miše
Ivan Nador
Radomir Nikolajev
Đorđe Janković
Dejan Ristanović
Zoran Obradović
Miodrag Potkonjak
Jelena Rupnik
Dušan Slavić
Nevenka Spalević
Jovan Skuljan
Srđan Stakić
Mihajlo Karapandžić

Naslovna strana
Vladimir Simović
Fotografije
Vladimir Simović
Ilustracije
Miodrag Marković

Prevodioci
Esad Jakupović
Ksenija Pješčić-Lebedinski
Izdavački savet „Galaksije"
Dr Rudi Debijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Gligić, Esad Jakupović, Velizar Masiać, Nikola Pajić, Željka Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vesović, Miloje Vuković
Štampa
Beogradsko izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17
Žiro-račun kod SDK 60802-633-2463
Devizni račun kod Beobanke
60811-620-6-82701-999-01066
Za inostranstvo cena dvostruka (400 D, 2,50 US $, 6.50 DM, 45 Sch, 5,50 Sfrs, 20 Ffrs).
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika" broj 26/72, ovo izdanje oslobođeno je poreza na promet

Šta ima novo

Dvoboj divova

Nismo se još ni privikli na postojanje jednog moćnog, a pristupačnog procesora kao što je MK68000, a proizvođač NATIONAL SEMICONDUCTOR se postarao da nas iznenadi novim, po ceni već pdatupačnim, tridesetdvobitnim procesorom

NS32332 je nastao kao logičan proizvod trke za prestiž između NATIONAL SEMICONDUCTOR-a i INTEL-a Intel je u početku ostvario premoć svojim brzim procesorom 80386 Ova procesor je sa svojih 3—4 miliona instrukcija po sekundi (MIPS) bio oko 4 puta brži od serije 32000 National Semiconductor-a Pojavom poboljšane verzije NS32332 ova trka za prestiž sasvim sigurno dobija potpuno nove okvire

Ovoga puta pred nama je nešto sasvim izuzetno Sićušno parče kristala obrađeno VLSI tehnologijom, u 84 pinskom pin-grid array pakovanju, može da izvrši 2 5 do 3 miliona operacija u sekundi Za adresiranje, pdatupačan mu je prostor od fantastičnih 4 G b (GIGA bajta) centralne memorije po zadatku NS32332 poseduje potpunu tridesetdvobitnu unutrašnju strukturu potpuno simetričan set instrukcija HLL (high level languages) instrukcije sa 4 različita HLL načina adresiranja, prave registre opšte namene i podršku aritmetike pokretnog zareza Proizvođač se, za razliku od dosadašnje prakse poztarao da zadrži potpunu kompatibilnost sa ranijom serijom, kako na gore tako i na dole To znači da će svi programi pisani za seriju 32000 raditi i na novoj seriji 32300 i obratno. Na novom procesoru može se koristiti sav softver razvijen za seriju 32000 Ovo podrazumeva mogućnost programiranja u fortranu, paskalu, C-u i donekle, u ADI, čija se implementacija za seriju upravo dovršava Od operativnih sistema dostupni su AT&T UNIX SYSTEM V Microsoft Corporation XENIX i Nacionalova verzija UNIX-a 4 1 GENIX Svi ovi operativni sistemi obezbeđuju pejdžovanje virtuelne memorije ' Pošto MMU hardverski rešava neke od ključnih funkcija rada se virtuelnom memorijom, objektni kod generisan kompajlerom, je za oko 20 posto kompaktniji nego kod računara VAX 11/750

Upotrebna vrednost novog mikroprocesora, velikih performansi je vrlo široka Već danas NS32000 i NS32332 sa svojim pratećim komponentama kodate se u CAD/CAE računarskim stanicama, za konstrukciju poslovnih računara u robotici, pri izgradnji multiprocesorskih sistema i tako dalje Obzirom na cenu koja sada iznosi oko 200 US dolara verovatno da nije daleko dan kada će mnoštvo „komodora", „spektruma" „atarija" ... biti zamenjeno mašinama po mogućnosti bliskim računaru tipa VAX 11/750

Radomir Nikolajev

Kucni IBM

Da li se IBM nadnosi nad tržišta kućnih računara? Možda i ne ali je zato dobrano prepiašio to tržište svojim IBM JX-om To je računar koji je pre poztojao samo u japanskoj verziji ali je sada prepravljen na englesku varijantu i za probu gurnut na australijsko tržište Neki tvrde da je to samo nova varijanta „PC juniora" dok drugi misle da se radi o dugo najavljivanom PC 2 Biće da ni jedni ni drugi nisu u pravu iako bi bilo interesantno videti šta zaista jeste taj računar Ništa kod IBM-a nije naivno

Robot operator

Radio-klub Nikola Tesla iz Beograda (Timočka 18) je poznat po prvoj kompjuterskoj sekciji u Jugoslaviji dok je 1979 godine slabo ko znao šta kućni računar uopšte može nezanemarljiv broj učenika beogradskih škola (računajući i autora ovoga teksta) imeo je prilike da oproba svoje (programarske) sposobnosti na „pravom pravcatom" kućnom računaru — TSR 80 I

Radio-klub Nikola Tesla je nedavno ostvario još jedan uspeh — primenio je računar u obučavanju mladih radio-amatera i tako je u saradnji sa Poslovnom zajednicom radio-difuznih organizacija Srbije nastao prvi robot-operator na amaterskim radio-stanicama u Jugoslaviji

Radio-amateri dobro znaju da je za sva takmičenja bitan broj održanih veza i daljina na koju su te veze održavane Da bi se ovi podaci registrovali na početku svakog razgovora treba razmeniti pozivne znakova dati izveštaj o kvalitetu prijema izračunati rastojanja i uraditi nekoliko drugih sitnica koje mladi telegrafisti redovno zaboravljaju Ukoliko se međutim zadesite na frekvenciji 144 075 MHz ili 3560 KHz dočekaće vas pozivni znak YU1R — prvi ameterski robot operator Robot će od vas zahtevati da se identifikujete i date druge protokolom predviđene podatke, a onda će vam uzvratiti ravnom merom Zatim će podaci o vezi biti upisani u računaravu evidenciju i dopunjeni rastojanjem između vas i kompjutera koje će biti jednostavno izračunato prema vašem pozivnom znaku Posle nekoliko dana ćete dobiti potvrdu o održanoj vezi koja će važiti za bilo koje takmičenje radio-amatera

Iako je sasvim jasno da robot-operator ne može da razgovara sa vama kao Daneel Olivawa tvorcu ovog projekta možemo da uputimo samo komplimenta — ing Zvonimir Makovec je našao izvanredan način za primenu kućnog računara Ukoliko želite da porazgovarate o njegovom delu pozivni znak ing Makoveca je YU1ZM dok je pozivni znak Radio-kluba Nikola Tesla YU-1AHI Pozivni znak robota je da ponovimo još jednom, YU1R

Dejan Ristanović

Univerzalna pisaća mašina

Ako ste od onih koji znaju puno jezika ili bar jedan sa malo egzotičnijim pismom ili na kraju krajeva bar ćirilicu i latinicu pa hoćete da koristite oba pisma u prepisci, onda je vaše probleme rešio jedan od brojnih univerziteta u Bostonu Brandeis univerzitet je realizovao prvu fazu softvera koji će omogućavati pisanje na 192 svetska jezika Ova faza obuhvata 16 jezika uključujući engleski ruski arapski i hebrajski Program je izuzetno koristan za lingvište kojima na primer omogućava istovremeno prevođenje i ispisivanje teksta na terminalu ili štampaču sa srednjovekovnog ruskog na savremeni ruski i engleski tako što se na ekranu terminala istovremeno prikazuju sva tri teksta Slova na tastaturi se ne menjaju pri promeni pisma već se na terminalu pokazuje koje slovo novog pisma odgovara tipki standardne tastature Pošto je ekran od tečnog kristala nema ni neugodnog treperenja pri dužem radu Sistem će se uskoro naći i u komercijalnoj prodaji i prvenstveno je namenjen političkim i komercijalnim organizacijama koje imaju intenzivne veze sa inostranstvom

N Aleksić

Šta ima novo

Proširenja RAM-a

Engleska firma Dk Tronics je ponudila tržištu RAM proširenja za Amstradove modele 464.664 i 6128 — preklapajuća RAM pakovanja i RAM disk

RAM pakovanja se isporučuju u dve varijante od 64 K i 256 K sa cenama od 49 95 i 99 95 funti Kao i kod CPC-a 6128 nije vam omogućeno pisanje dužih bejzik programa ali zato dodatnu memoriju možete iskoristiti za smeštanja podataka ostavljajući time više mesta za program u glavnom RAM-u Prateći softver uvodi desetak novih bejzik naredbi za manipulisanja i pristup ovom proširenju memorije iako je preklapanja ovih RAM blokova moguće i bez dodatnog softvera Ovo proširenja omogućava — vlasnicima CPC-a 464 i 664 da rade sa programima pisanim i za CP/M plus

Druga novost je efficijumski RAM disk od 256 K koji se može koristiti kao drugi ili treći disk drajv Najveća prednost RAM diska je njegova brzina jer će se program daleko brže izvršavati nego prilikom rada sa normalnim diskom Upravljanja je moguća iz bajzika dodatnim RSX naredbama DISCLOAD i DISCSAVE kao i pod CP/M-om uz pomoć SETUP COM programa Sadržaj RAM diska kao i memorijskih proširenja, ostaje sačuvan i nakon resetovanja računara ali ne i nakon isključenja Cena od 99 95 funti je dosta umerena ali ne treba zaboraviti da se Amstradova druga normalna disk jedinica može nabaviti po istoj ceni

D Greber

Programi za pisce

Pojavljuju se i prvi programi za Amstradov PCW 8256 iako su neki zlobnici tvrdili da je to možda zgodan aparat za obradu teksta ali da nikad neće biti neki naročit računar Radi se o tekst procesorima i komunikacijskim paketima

Više memorije za PCW 8256

Još malo o PCW 8256 Ako se razumete u elektroniku možete da otvorite kućište računara i tamo ćete otkriti prazna podnožja za memorijske čipove izgleda da je Amstrad nameravao da koristi više čipova sa manjom memorijom, ali je u toku završavanja računara došlo do naglog pojeftinjenja kapacitetnijih čipova pa su iskorišćeni oni i zato je ostalo mesta za još nekoliko čipova Tako ako budete pažljivi u rukovanju iznutricama računara možete proširiti memoriju „PCW 8256" za 256K Nije malo

Na niske grane

Cirtech je izbacio pločicu sa Z-80 koja računaru „epl ttc" omogućava da se na njemu koristi CP/M Plus Izuzetno značajan napredak! Da li to znači da možemo uskoro očekivati da Eplovi računari počnu da imitiraju čak i „komodor 64" i „spektrum"

Unix za slepe

Britanski Royal National Institute počeo je obimni program čiji je cilj omogućavanje slepim osobama da u što većem obimu koriste mogućnosti računara Prvi projekat je organizacija petodnevnog Unix kursa za slepe Takođe je prevedena na Brajovu azbuku i knjiga Banahana i Ruttera „Unix the book" Udžbenik koji je takođe štampan izveden je u mašinski čitljivom obliku tako da se lako može pomoću specijalnih mašina prebaciti u Brajovu azbuku umesto mukotrpnog skupog i sporog posla prebacivanja karakter po karakter iz klasičnog formata

N Aleksić

Crtanje na QL-u

Softverska kompanija PSION za koju se u poslednje vreme ne čuje mnogo a koja se proslavila odličnim programima za QL, pustila je u prodaju jedan novi program za QL računar Program se zove „Q DRAW" (gde je ostalo slovo L?) a služi za crtanje Taj isti program su upotrebljavali programeri ove firme za kreiranje grafike za odlično ocenjene programa „QL Chess" (šahovski program) i „Match Point Sada je dostupan svakom po ceni od 15 funti

Čipovi preko satelita

Intel je naručio izgradnju instalacije u San Huanu u Portoriku koja će omogućiti naručivanje bilo kog Intelovog čipa koji je proizveden u bilo kom delu sveta preko satelita za kupce širom sveta Instalacije izgrađuju britanska firma Litton po cent od 10 miliona dolara Pored informacija koje interesuju kupce Intel će i sve podatke koji omogućavaju efikasnu i ekonomičnu proizvodnju na primer o zalihama sirovina potrebnim za proizvodnju prenositi preko satelita do svog sedišta u Santa Klari u Kaliforniji Tako će kompanija imati sve potrebne podatke o bilo kom od svojih 16200 proizvoda

N Aleksić

Poklon za igru

Cena „komodora 16" tako naglo pada da se može očekivati da će ga uskoro davati kao poklon uz svaku igru za njega koju kupite

Piratski ćale

Izgleda da se Amstrad izvukao iz susedskog procesa koji se protiv njega vodi zato što je svoj dupli kasetaš reklamirao kao idealnu stvarčicu za pirate Sud je odlučio da nije nezakonito prodavati takvu opremu ali da ne sme u reklami da se potiče na piratstvo

Za mentalno zaostale

Na univerzitetu Kil (Keele University) POD imenom Mr Ugh je napravljena prva video igra za mentalno zaostale Zlobnici bi rekli da su sve takve

Ovde daleko

Izgleda da Britanska Francuska i Italijanska vlade povoljno gledaju na pokušaj Ejkorna, Tompsona i Olivetija da razviju evropski standard za obrazovni računar Kad se zna kako je Ejkorn uspeo sa BBC-jem, to i nije tako iluzorno Osim možda za nas, jer mi smo daleko daleko

B Đak

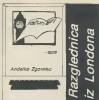
Kvalitet i jeftinoća

Saga Sistems (Saga Systems) polako dokazuje da je jedan od najozbiljnijih proizvođača za spektrum Najsvežiji primer je veoma dobar štampač pod oznakom LTR-1 Štampač pruža otisak sasvim nalik pisaćoj mašini, iako je izuzetno jeftin — košta samo 149 funti (možda bi ovo „izuzetno jeftin" za naše prilike trebalo pretvoriti u „relativno jeftin") Printer prima papir formata A4 u odvojenim komadima Tih nije — ali nije nešto naročito ni bučan Sa bidirekcionalnim štampanjem dostiže brzinu od 10 znakova u sekundi Za kraj još treba reći da ima standardan „centronika" interfejs

B Đak

Tajni greh

Lotus je javno priznao da je njihov polulegendarni program Simfoni 1 1 (Symphony 1 1) ima bubicu (bag) koja može u trenu da pojede veliku količinu informacija Do trenutka njihovog priznanja bilo je prodato preko 25 000 komada programa i veliki greše A tek mali



Anđelko Zgorelec

Razglednica iz Londona

Žir iz pepela

Protekla je još jedna računarska godina — godina u kojoj smo slavili desetogodišnjicu mikroračunarstva, ali i godina koja je bila i veoma teška za ovu mladu industriju. Jedni proizvođači su propali, drugi su se jedva održali. No, izgleda da je ono najgore prošlo i mikroračunari su opet počeli da se prodaju. Proizvođači i trgovačka mreža nisu baš najsrećniji jer su morali drastično da snize cene za neke modele čak i napola za godinu dana, ali glavno da je krenulo. Ljubitelji mikroračunarstva se sigurno ne bune — uštedeće bar malo novaca ako kupe nešto starije modele, a i novim računarima, koji imaju odlične specifikacije, cena će biti znatno manja. Interes u Britaniji za mikroračunare je ponovo toliko porastao da su neke radnje sigurno zažalile što su uklonile računare sa svojih polica. Iz najveće trgovačke mreže za prodaju elektronskih produkata Dixon javljaju da prodaja ide vrlo dobro. Isto predviđa i vodeći poslovni dnevnik „Fajnenšel Tajms" (Financial Times) koji smatra da će prodaja računara opet krenuti nabolje.

Računari u paketu

Predstavnik najvećeg nezavisnog distributera za Amstradove i Komodorove računare, firma Lajtning (Lightning), Loretta Cohen smatra da je tokom prošle godine u Britaniji prodato preko milion malih računara, što je veoma ohrabrujuća brojka. Za razliku od prošle godine, trgovačka mreža više ne govori o prodatim računarima kao jedinici, već o takozvanim „paketima" — računar sa nekim periferním dodacima poput kasetofona, palice, pa čak i disk jedinice, prodaje se u „paketu" znatno jeftinije.

Nema sumnje da su potencijalni kupci zainteresirani za „pakete" jer sada im je potreban samo televizor da računar bude u punoj spremi. Na taj se način takođe smanjuje i mogućnost greške da se kupi nešto što nije kompatibilno sa samim računarom.

Mnogi modeli računara se sada prodaju po „damping" cenama, pa neki trgovci tvrde da računari neće biti nikad više tako jeftini. Ima modela računara čija je cena sada samo 29 funti („spectravideo"), a do nedavno su se prodavali i za preko 200 funti. Ne bismo međutim, nikome savetovali da kupi neki od takvih računara jer za njih gotovo nema programska podrška niti ima izgleda da će, pošto se više ne proizvode, za njih nabaviti rezervni delovi. Dakle, ova „povoljna kupovina", ili kako se to ovde kaže „bargain", je samo za — najhrabrije. Nešto bolja situacija je da se kupi neki od Komodorovih računara (modeli Plus/4 ili C16) jer ima nešto više programa, ali ni taj izbor ne bismo preporučili jer i to su već „otpisani" računari koje Komodor više neće proizvoditi, pa se ni jednoj softverskoj kompaniji ne isplati da za njih piše softver (Komodor sada koncentriše sve napore na računare 128 i „amigu").

U modi je „amstrad"

Šta onda, dakle, kupiti? Vaš dopisnik preporučuje svojim prijateljima i poznanicima koji dolaze u London (jasno ako im je potreban savet) da nabave neki od računara koji ima veliku korisničku bazu u našoj zemlji — spektrum, komodor 64, „apl" ili neki od „amstrada". To su provereni računari, sa dosta softvera kod nas i velikim krugom korisnika.

Najnoviji Amstradov proizvod PCW8256 dobio je neobično velik publicitet (pored ostalog i naslovna strana časopisa „Personal Computer World") jer je to kompletan računarski sistem — monitor, štampač i jedna disk jedinica — koji se prodaje po ceni od samo 399 funti bez poreza. Računar se, izgleda, definitivno može preporučiti za one koji mnogo pišu, kao i za malu poslovnu primenu. Računar se za sada odlično prodaje i gotovo ga je nemoguće nabaviti i pored mesečne produkcije od 40 000 primeraka iz nekoliko fabrika u Južnoj Koreji. Naročito je popularan u poslovnim krugovima većih firmi jer kako tvrdi jedan list, šefovima poslovnih jedinica nije potrebno odobrenje odozgo za nabavke do 500 funti, pa se kupovina ovog računara jednostavno zavodi u poslovne knjige kao nabavka sitnog kancelarijskog pribora. Pošto vaš dopisnik upotrebljava svoje računare najviše za obradu teksta i datoteku (navodno čak 70% vlasnika računara ih koristi u iste svrhe), za njega je PCW8256 računar godine.



Ejkorn „komunikator"

Sredinom novembra se već tradicionalno održava najveći kompjuterski sajam u Britaniji zvani Compec. Na njemu je prethodnih godina bilo najviše velikih i miniračunara, dok je ove godine bilo izloženo oko 100 različitih modela mikroračunara među kojima su neki pobudili veliki interes. Zvezda sajma međutim nije bio mikroračunar neke od profesionalnih firmi, već dolazi od firme Ejkorn (Acorn = žir) koja je jedva preživela ovu godinu (Ejkorn nikako da se reši problema, najnoviji je došao od poreskih organa koji tvrde da im ova kompanija duguje zaostali porez u iznosu od 2,8 miliona funti za 1983. godinu).

Računar je nazvan Acron Communicator i to je prvi računar ove firme sa 16-bitnim procesorom WD65SC816 koji je potpuno kompatibilan sa dobro poznatim 6502 koji pored Acorna za svoje računare koriste i Apple, Atari i Commodore. Ali ovaj procesor ima 24-bitni adresni bas, pa korisnička memorija nije više ograničena na 64 K, već se poput Motorolinog procesora 68000 može kretati do gigantskih 16 megabajta.

Communicator je zapravo mali poslovni kancelarijski računar sa ugrađenim modemom i telefonom, ali se na njega mogu priključiti disk jedinice i ostali dodaci od toliko popularnog BBC računara, kao i ROM čipovi sa programima. U ovaj računar je ugrađen i Econet, pa je moguće povezati čak 254 računara u mrežu, što je velika prednost za poslovnu upotrebu. Cena se za sada još ne zna, ali će biti oko 1000 funti.

Takozvani dobro obavešteni krugovi smatraju da će ovaj računar biti baza za najnoviji model BBC-jevog 16-bitnog računara, koji bi trebalo da se pojavi u prodaji u prvoj polovini ove godine. Izgleda da se Ejkorn nada da bi takav računar mogao da bude usvojen kao standard za evropski obrazovni računar.

Šta ima novo

CP/M na „amstradu"

Za vlasnike „amstrada" pojavio se veliki broj prebačenih i prilagođenih CP/M programa na formatu od 3 inča.

Najinteresantniji od njih su svakako programi poznate firme Digital Research — PASCAL/MT+ i CBASIC COMPILER, kao i grafički paketi DR DRAW i DR GRAPH. Za svaki od njih treba izdvojiti 49.95 funti.

Oxford Computer Systems je takođe izbacila na tržište svoju verziju paskal kompajlera (OXFORD PASCAL) za modele 6128 i 8256 po ceni od 49.95 funti.

Firma MicroPro je prilagodila najpoznatiji tekst-program, sada pod imenom POCKET WORD STAR, ali ne i džepnom cenom od 119 funti. Program zahteva najmanje 56 K memorije, tako da je kompatibilan već sa CP/M 2.0. Uskoro će biti predstavljeni POCKET CALCSTAR (spreadsheet program) i POCKET DATASTAR (baza podataka).

Za one koji se bave vođenjem računovodstva i obradom platnih spiskova, programska kuća Sage im je obezbedila vrhunske programe (po vrhunskim cenama) i to POPULAR ACCOUNTS (100) funti) i POPULAR PAYROLL (70 funti).

Ovo svakako nije konačan spisak noviteta, jer se pretpostavlja da je amstradova biblioteka CP/M programa već dostigla brojku od dve stotine.

D. Graber

Prstom po ekranu

Da li ste ikada poželeli da prstom pišete po ekranu računara? Ako jeste, ovo vam je idealna prilika da to i probate. Britanska firma Barnard Elektroniks isporučuje paket pod imenom Goust Skrin (Ghost Screen) koji vam omogućava da brljate direktno prstom po ekranu. Paket sadrži monitor, demonstracioni softver i korisnički paket softvera za ubacivanje u vaše sopstvene programa. Postoje verzije za „spektrum" i „komodor 64". Kada spojite aparaturu sa vašim računarom, pažljivo pročitate uputstva i učitajte program. Dodirujući prstom ekran možete na njemu da ostavljate trag u obliku linija. Zar ne izgleda mnogo bolje od svetlosne olovke? Cena takvog paketa je oko 150 funti.

Jagma za disketama

Na računarskom tržištu oseća se veliki nedostatak disketa od 3 inča. Izgleda da su i proizvođači, i distributeri i prodavci potpuno zakazali u proceni prodaje Amstradovih računara. Ništa, naučiće sa za sledeći put.

Na udaru kritike

Izgleda da je Atarijev „ST 520" najzad završen. Bio je on i pre skoro gotov, osim što je malo kuburio u vezi sa operativnim sistemom. Samo to. Sad je gotov i već stižu prve zamerke. Zamerke se tiču super igre za njega Bratakasa, za koju oni koji su je probali masovno tvrde da predstavlja direktnu kopiju Megaigre za „komodor". Ko zna, možda su i u pravu. Čućemo šta Atari ima da kaže na to.

Jeftinija „galaksija"

Dok pripremamo ovaj broj saznajemo da će se računar „Galaksija" do 31. decembra prodavati po izuzetnoj ceni od 55.555 dinara. Mi ćemo izaći iz štampe tek kad taj popust prođe, ali nismo mogli da propustimo a da ne pomenemo tu neverovatnu vest. Da li je neko pomenuo krizu?

Taksi za PC

Epsonov programski paket „Taxi", napisan za Epson PC, koji je omogućavao sve blagodeti koje ima „mekintoš", i to u koloru, sada se može dobiti i u varijanti za original PC. To je veliki plus, jer je TAXI zaista izvrstan program koji omogućava da čak i druge programe koje koristite na PCju obogatite njegovim mogućnostima, što nije odlika, recimo, GEMa.

Listing na majici

U prodaji su specijalne printerske trake koje omogućuju takvu vrstu štampe kakva se posle pomoću vruće pegle može preneti na majicu koja ima najmanje 35% poliestera. Da li ste razmišljali da malo švrljate po vašoj majici pomoću računara?

„Epson" kao „epl"

Finger print je set od tri dopunska čipa za Epsonovu FX-80 seriju koji čine da „epson" štampa nalik IBM-ovom i eplovom (Apple) printeru. To nije sve. Moguć je i NLQ mod koji je jako sličan printeru sa lepezom.

Nožni miš

Jedno veliko [illegible] u hardverskim dodacima za računare! Radi se o mišu, ali ne o običnom, nego o nožnom! To je uređaj za IBM P6 koji se drži ispod stola i pomera nogom. Funkcija mu je ista kao i normalnom „stonom" mišu, a izgleda kao metalna kutijica sa gumenom papučicom. Sad se, dakle, računarima može raditi i pomoću nogu. Šta je sledeće?

Grafički paketi

Posle tekst procesora i spredšitova, evo još jedna udarna tema za softversku industriju kućnih računara. U prodaji je sve više grafičkih programskih paketa za crtkanje i bojadisanje. Najpopularniji su Art studio (IBM PC), Fluent Fonts (Mac), Mac the Knife (Mac), Dotwriter 4.0 (Apple II, Mac, C 64, Atari 400—800), Blazing Paddles (Apple II, C 64) i PC Pallete (IBM PC). Sigurno će ih biti još, sve boljih i sve jeftinijih.

Muzej kompjutera

Ako ste od onih koji imaju sve osim čokoladnog kompjuterskog čipa ili releа koji se koristio za prvi računar na Harvardu, onda nema problema — pišite kompjuterskom muzeju na adresu: The Computer Museum Store, 300 Congress Street, Boston, MA02210 USA i oni će vam poslati besplatan katalog. Kompjuterski muzej prati razvoj procesa obrade informacije od računaljke do sićušnih i galijum-arsenid čipova putem izložbi, kataloga, knjiga i programa. Ovo je jedini muzej ove vrste u svetu, bar za sad.

N. Aleksić

Load „dragi računar"

Drugarska pretnja

Ako u roku od mesec dana ne objavite bar jedno od šest pisama koje sam vam poslao prestaću da vam ih pišem! Najozbiljnije!

Velibor Tederavić, Zemun

Dragi Velibore, javi nam se telefonom. Želeli bismo da se upoznamo sa tobom.

Viša praksa

U nekim brojevima Vaše revije pisali ste o raznim primenama računara. Mišljenja sam da bi trebalo odvojiti više prostora za pisanje o upotrebi računara u oblastima prirodnih nauka, arhitekturi, građevinarstvu itd. Pišite malo više o aplikacijama u hemijskoj, fizičkoj, elektronskoj i drugim laboratorijama. Bilo bi lepo kad bi sve to bilo propraćeno i prigodnim programima iz konkretnih oblasti. Možda ne bi bilo loše kad bi se povremeno pisalo i o specijalizovanim računarima.

O. Spiroveki, Kumanovo

Kome da verujem

Voleo bih da mi razjasnite jednu stvar koja me čudi. Kada sam prvi put čuo za Amstradov „PCW 8256" bio sam veoma zainteresovan jer je svoj računar uglavnom koristim baš za obradu teksta. Zato je razumljivo što sam se puno obradovao kada sam video da u „Računarima 10" imate pregled te mašine. U stvari, nije bio jedan pregled već dva jer su i Dejan Ristanović i Branko Đaković pisali o računaru „PCW 8256". Ono što me čudi je to što su njihova ocena nepomirljivo različite. Voleo bih da znam da li se to dogodilo slučajno ili namerno i kako da tumačim takve ocene. Verni čitalac

Miloš Jeftović, Beograd

Da si čitao pisma u „Računarima 9" primetio bi da je već rečeno da ćemo pokušati da za svaku novu mašinu bude bar dva pregleda ili pogleda, radi objektivnosti i sadržajnosti. U ovom slučaju Dejan je gledao sa aspekta računarskog stručnjaka a Branko samo kao korisnik. Kao i svaki drugi računar i „PCW 8256" ima dva lica. Trudićemo se da i dalje na takav način predstavljamo nove mašine.

Knjiga od „Računara"

Imam još od pre jednu ideju koja mi se ponovo javila kad sam video vaš registar tema o kojima ste pisali u dosadašnjim brojevima „Računara". Znam vaš običaj da na kraju godine ukoričite izvestan broj. Galaksija i da ih tako prodajete. Mislim da bi slična ideja sa „Računarima" bila izvanredna. Takva knjiga bi predstavljala zaista dragocenu pomoć za bilo kog pasioniranog programera. Možda bi jedini problem mogla da predstavlja činjenica da je prva dva broja „Računara" rasprodato ali vi biste to mogli da doštampate, zar ne? Kad se takva „knjiga" pojavi, ako se pojavi, ja ću biti siguran kupac iako imam sve dosadašnje „Računare".

Goran Janković, Banja Luka

Ispunjenje snova

Bilo mi je veoma drago kada sam u poslednjem broju „Računara" pročitao članak o gradnji takozvanog kertridža za C-64. Veoma bih voleo da učestvujem u realizaciji ove super-ideje a za početak dajem nekoliko predloga za programe koji bi bili na kertridžu: 1. Simon's Basic 2. Hesb 3. Monitor 49152 ili slično. Ideja o kertridžu u koji bi bila ugrađena rutina za korišćenje grafike (i muzike) promenom editora je dobra. Međutim to mi više liči na promenu ROM-a a o tome treba posebno pričati. Zbog lošeg vida bio sam prinuđen da napustim i elektroniku i računare (po savetu doktora) ali sam u ovom članku video ostvarenje svojih snova.

Radeta Nebojša
Jovana Cvijića 7/6
15000 Šabac

Zanemarena klasa

Nepravedno ste zanemarili cijelu klasu džepnih računara. Uprkos očiglednim manama, to nisu samo igračke. Ne sećam se da sam išta pročitao o modelu SHARP 5100 kojim se služim već pet godina i koji u potpunosti zadovoljava moje sitnije svakodnevne potrebe iako se na njemu ne mogu ubijati mrski osvajači iz svemira. Nedavno sam u reviji „Naš Dom" čitao o vrlo interesantnim modelima džepnih računala. Nadam se da ni „Računarima" neće biti ispod časti da objave nešto o takvim mašinama, o njihovoj primjeni i pripadajućoj opremi. Puno pozdrava.

B. Š. Zagreb

Ne sećamo se da poznajemo ikoga ko koristi sharp 1500. Zašto ne pripremiš nešto za svoje istomišljenike?

Muke sa šiframa

Na rešavanje vašeg nagradnog pitanja broj 2 utrošio sam oko dvanaest časova i nisam uopšte koristio računar. Karaktere sam prebrojavao ručno jer bi mi za taj posao računarom bilo potrebno gotovo isto toliko vremena.

Kao dokaz da sam tekst dešifrovao, evo prve dve rečenice teksta: Ja se zovem Robin Brodhed i najbogatija sam osoba u čitavom sunčevom sistemu.

Posle dešifrovanja teksta hteo sam da uradim isto ono što je i Dejan uradio. Za to mi je trebao broj N u RANDOMIZE N. Da bih ga dobio modifikovao sam Dejanov program za šifrovanje (zato što moj spektrum nema funkciju MID$) ubacio ga u FOR NEXT petlju i zapetljao računar u posao s kojim se ovaj bono čitavo popodne a stigao je da ispita samo prvih 3000 slučajeva (od 65535). Zato sam odustao.

Želim redakciji da nastavi putem kojim je krenula!

Tihomir Kučinar

Ni luk jeo, ni luk mirisao

„Računare" smatram izvesnim časopisom ali su me u poslednje vreme malo razočarali. Čini mi se da „Računari" kao uostalom i ostali računarski časopisi zaboravljaju činjenicu da je tipičan YU haker srednjoškolac ili student. Zbog toga mu je potpuno svejedno da li amiga košta 1000 ili 100 000 dolara! Bilo bi nepravedno ne pisati i o takvim računarima ali je još nepravednije sve posmatrati kroz njih! Smatram da je suviše vređanje YU hakera po sistemu — spektrume, komodore i QL-ove bacite u smeće, tu je amiga. Ovi računari još dugo neće kod nas u smeće, pogotovo ako dinar nastavi ovako padati. Mislim da vas je i vaša iskustva uverila u to jer bolje je vrabac u ruci...

Druga stvar koja mi smeta a smatram je i puno opasnijom jeste neobjektivnost. Ova zamerka se najviše odnosi na tekstove Dejana Ristanovića.

Citiraću: — „Spuštanje cene računara na 200 funti ... ovaj potez je obična potvrda više puta izrečene tvrdnje da je QL promašena mašina" (broj 8, strana 4, naslov „Nikada ne veruj Sinkleru").

— „Rad sa mikrodrajvom je toliko mukotrpan da se mnogi vlasnici spektruma bolje osećaju bez njega nego sa njim" (broj 4, strana 26).

*— „Očekuj uskoro **jedan** dobar članak o QL-u. Neki zlobnici doduše pokušavaju da u domaći folklor provuku izreku „Veruj, keru ne veruj Sinkleru", **ali mi ne spadamo među njih**" (broj 10, strana 19).*

Ovo zadnje doduše nije potpisano ali je slično naslovu prvog citata a i po stilu liči na Dejana.

Ovakvi naizgled sitni propusti opravdano izazivaju revolt kod čitalaca.

Umeci su vam pun pogodak.

Ne bih vam ovo pisao da ne volim „Računare". P. S. Dejana izgrdite. Vladu pozdravite a Jelenu poljubite u moje ime!

Mihaličke Damir
Gundulićeva 148
54000 Osijek

Vladu smo pozdravili, Jelenu smo poljubili a Dejana, naravno, nismo izgrdili. On nama nikakve veze sa onim što nam zameraš. Neka se sam javi ko je kriv! P. S. Zamoli nas još jednom ono poslednje pod P. S.

specijalno izdanje rubrike load „dragi računari" objavljujemo na str. 66

Boris o sebi

Evo ponovo vam se javljam da bih dopunio svoje pismo koje ste objavili u „Računarima 10". Zamenili ste mi što vam nisam napisao potpunu adresu i prezime. Sada vam ih šaljem kao poklon za Novu 1986. godinu. Zovem se Vulović Boris, stanujem u ulici D. Vukasovića 78, blok 61a, Novi Beograd.

Boris iz Devete beogradske

P. S. Najbolji ste. ŽIVEO QL!

Drugi o Borisu

U „Računarima 10" sam pročitao pismo „Borisa iz Devete beogradske" (idem u isto odeljenje sa njim). Prvo bih želeo da ispravim njegovu grešku i objavim njegovu adresu: Vulović Boris, Dušana Vukasovića 78/14, 11070 Novi Beograd.

Mislim da je Boris u svom pismu preterao sa hvaljenjem QL-a. Sigurno ste primetili da Boris nije naveo ni jednu njegovu manu. Piše da tastatura zadovoljava. Ako pritisnete tipku tačno u sredinu nema problema ali ako je pritisnete bliže ivici tipka koči i otežava kucanje.

Softver — najkvalitetniji. Uostalom kad je pokazao meni zašto čitaocima „Računara" nije napisao i cene programa za QL? Zato što bi nekoliko programa zajedno prevazišli cenu QL-a!

Da li je QL-ov editor koji je ostao na nivou spektruma dobar? Naravno da nije! Zašto su napravili QL Toolkit koji pokušava da od QL-ovog bejzika napravi dobar bejzik?

Boris je želeo da kupi QL. Tada je isticao samo njegove dobre osobine a za mane se pravio glup. Onda se naivno odlučio da kupi atari 520 ST i naravno isticao samo dobre osobine a QL-u je isticao samo mane (mikrodrajv je najveća glupost na svetu, tastatura nije profesionalna, editor je glup). Kad je otkrio da „atari" ne odgovara njegovim potrebama vratio se QL-u zaboravivši da je nekada pričao o njegovim manama. Počeo je ponovo da ga hvali i diže do neba njegov procesor 68008. Jedan njegov drug misli da je oboleo od nove bolesti QLTITISA. Simptomi su: oboleli više od 100 puta u toku dana izgovori reč QL.

Ja nemam ništa protiv QL-a ali sam protiv preuveličavanja njegovih dobrih karakteristika.

Čelenković Slobodan
(uz pomoć Trifunj Zlatke)
iz Devete beogradske

Drage računardžije iz Devete beogradske, lačno je da ste lepo prikazali i zagrižene pobornike QL-a i njegove kritizere ali zar ne smatrate da bi se ovaj novinski razgovor mogao nastaviti u miru prelepe Devete beogradske? Što se Borisa tiče, svako ima pravo i da nešto odluči i da se — predomisli.

Priprema: Branko Đaković

računar po glavi

Kako to radimo mi

Posle čitanja ovakvog uvoda prosečnom čitaocu neupućenom u filozofiju našeg računarskog biznisa sasvim sigurno zastaje dah. Niko od nas nije zamišljao da put u vode zdrave ekonomije može biti tako jednostavan. No već ovde sasvim sigurno postavljaju se i neka pitanja na koja nije lako dati odgovor. Nismo li mi negde na začelju po broju stručnjaka obučenih da se uhvate ukoštac sa problemima nove tehnologije? Nije li naša zemlja negde na začelju evropske lestvice tehnologije? Samo u dve naše najveće fabrike moguće je dobiti kvalitetnu dvoslojnu štampanu ploču. O višeslojnoj zasad nema ni pomena. Da li je osim toga neko čuo kako je neka naša fabrika uspela da proizvede makar i najbedniji mikroprocesor neophodan u računarskoj industriji? I kako se uopšte naša industrija bori sa ovim gotovo nerešivim problemima?

Veliki mali pirat

Kao i svuda u svetu i kod nas se računarska industrija deli na softversku i hardversku ali su veze između ova dva vida jače nego bilo gde drugde. Kod nas se prodajom softvera u pravom smislu reči bave jedino industrija hardvera i, naravno, nezaobilazni pirati. Ako se odlučite za nabavku računara nekog od naših proizvođača, on će vam u cenu proizvoda zaračunati i vrednost softvera koji se uz njega isporučuje. Za sve ostale softverske dodatke koji će vam u budućnosti trebati moraćete dobro da platite. Drugo rešenje je da se za ovakve softverske dodatke obratite piratima. Oni ih sasvim sigurno imaju i to po nižoj ceni. Manji problemi nastaju ako ste nabavili neki od većih računara (VAX PDP IBM HONEYWELL). Za ove računare još uvek nećete naći oglas u novinama koji nudi prodaju softvera ali ne očajavajte. Potražite nekoga ko ima takav računar duže od vas. On se gotovo sigurno već pozabavio sveopštom razmenom pameti.

Ako ne računamo minimalne zarade koje na ovakav način ostvaruju industrija hardvera i pirati, onda se ovaj vid sticanja računarskog hleba može gotovo sasvim zanemariti. Marginalnost ovakvog prilaza računarskom biznisu posledica je psihologija našeg društva. Mere za vrednovanje softverskih proizvoda kod nas još uvek nisu sasvim usaglašene. Još uvek se vode maratonske debate da li to treba da bude kilobajt čistog mašinca, jezik u kojem je program pisan ili pak neka sasvim nova mera koja će nepogrešivo utvrđivati vrednost svakog napisanog programa. Direktna posledica ovakvog pristupa je da kod nas u stvari čisto softverske kuće i ne postoje.

Drugi i daleko efektniji način prodaje softvera je posredni. Pri tom je potrebno da nabavite što veći računar. Ako uspete da uvezete neki CRAY ili nešto slično, bićete verovatno najjači u posrednoj prodaji softvera kod nas. Drugi korak u ovom poslu je da dovedete nekoliko stručnjaka (ne moraju biti pravi) sa što je moguće zvučnijim imenima. Na kraju, zbog vrednosti opreme i stručnosti kadra, proglašavate se računarskim centrom od opšteg značaja. Posle ovoga možete sasvim biti sigurni da vam je budućnost obezbeđena. Sve firme iz vaše mesne zajednice biće vam verni potrošači. Obrađivaćete im podatke dugo godina. Vaš računar će postati neizbežna banka podataka od koje zavisi čitav vaš kraj. I, naravno, eto neiscrpnog izvora zarade! Vi možete da diktirate cene, bez ikakvog ograničenja. Ako se neko pobuni, nema obrade, nema ni podataka. Naravno, ovde se kao jedini valjani savetnik pri kupovini računara morate neprestano truditi da suzbijete konkurenciju. Nekim od firmi za koje radite sasvim sigurno će pasti na pamet da i same osnuju slične centre. Tada vi nastupate kao savetnik i u njihovo ime kupujete neku bezvrednu mašinu. Najbolje da ta mašina može da obezbedi unos podataka ali nikako i obradu. Na taj način rešavate se dosadnog kucanja i viška skupe radne snage. To za vaš račun i smešno niska cenu rada svi ostali. Vi prihvatate ove podatke i obrađujete ih na vašem moćnom računaru. Naravno, u krajnju cenu obrade uračunavate sve: i vrednost amortizacije sistema i softvera, i unos koji ionako ne radite i ko zna šta još ne! Ljudska mašta je neiscrpna. Nemojte samo zaboraviti da na kraju kalkulisanja ovakve cene stavite prst u uvo i odoka pomnožite dobijene rezultate bar sa pet. Ostali koji se bave ovim poslom neće vam nikad oprostiti.

Uz svaku dlaku

Nedostaci ovakvog načina bavljenja računarskom industrijom su u visokim početnim ulaganjima. Cena mašine, prostora u koji će biti smeštena, papira neophodnih za otvaranje ovakve firme je od prilike nekoliko stotina milijardi (starih naravno). Osim toga ne treba zanemariti ni cenu stručnjaka neophodnih u ovoj igri. Oni danas savršeno znaju svoju vrednost i polako postaju ono što su nekad bili fudbaleri. A ako ih slučajno uvredite nekim vašim smešnim predlogom, odoše u Ameriku i — eto kraja vašem postu!

Drugi i daleko primamljiviji vid računarskog buma kod nas je proizvodnja hardvera. Istorijski gledano, u ovom našem večito burnom podneblju materijalne stvari bile su uvek cenjenije od duhovnih vrednosti. Tako se i danas opaža jedan na prvi pogled čudan paradoks. Dok u celom svetu cena hardvera drastično opada, a cena softvera nezadrživo raste, kod nas se zbiva sasvim obratan proces. Cena hardveru nezadrživo raste. Množe se komplikacije oko nabavke. Administracija uporno uzima svoj ne tako mali dinar pri svakoj kupovini mašina. Sve to polako počinje da liči na priču o duvanu. Čoveku je najlakše uzeti tamo gde je primoran da potroši. U isto vreme ta ista administracija još uvek je nezainteresovana za softver, pa njegova cena diktirana piratskim tržištem naglo opada. Ovakvi i na prvi pogled sasvim čudni uslovi na našem tržištu dali su snažan podsticaj savremenoj računarskoj tehnologiji kod nas. Sve je više promućurnih ljudi koji zaključuju da se igra u ovako mutnim vodama isplati. Adekvatan je i broj novonastalih firmi koje se bave proizvodnjom računara.

Ako živite u maloj nerazvijenoj opštini, možda je vreme da počnete da razmišljate o svemu što ste ovde pročitali. Možda je sve ovo pravi put da postanete spasitelj vašeg kraja. Prosvetitelj u novoj tehnološkoj revoluciji. Za sve to vam u stvari ne treba toliko puno kao što vam se možda čini. Za početak vašoj firmi je dovoljno nekoliko hiljada maraka da biste obezbedili zakonsku osnovu za uvoz. Neophodna su vam i dva-tri političara uticajnija u vašem kraju. Razgovarajte sa njima. Navedite im sve aspekte vezane za novu industriju i oni se neće oglušiti o vaš poziv. Oni veoma brzo računaju i biće im vrlo brzo sve ovo jasnije nego vama. Uskoro ćete imati sve saglasnosti neophodne za stvaranje nove fabrike računara. Odavde, naravno, sve zavisi samo od vas. Kada proizvodnja na kraju počne, tržište vam je sasvim sigurno obezbeđeno. Svi koji u vašem kraju budu želeli da nabave neku od inteligentnih mašina moraće da se obrate vama. To je u interesu regiona! Zato požurite. Drugi imaju slične ideje, a što pre počnete, veće tržište je pred vama.

Pljuni pa zalepi

Ako vas još uvek brine tehnologija proizvodnje, iznenadićete se koliko je to u stvari jednostavno. Vlada Butatović će mi, nadam se, oprostiti što ću se ovde poslužiti jednom njegovom pričicom. Verujem da će mu se dopasti otkriće da je on bio jedan od prvih vizionara naše računarske strategije.

Kod nas postoje gotovo isključivo dva metoda za proizvodnju računara. U oba slučaja postupak počinje na isti način. Prvo uvezete kompletan računar, najbolje iz Amerike. Zatim uvezete nalepnice, najbolje iz Italije. Pri tome morate smisliti što zvučnije ime novog proizvoda. Ako ste malo štedljiviji, nalepnice možete naručiti i na našem tržištu ali rede rizikujete da svi zaključe kako ste neozbiljan proizvođač i napuste vas. Čak je i naša tehnologija nalepnica zaostala.

Načelno, možemo razlikovati dve proizvodna postupka. U prvom od njih pljunete na nalepnicu, zalepite je na računar i prodate ga. Po drugoj tehnologiji prvo pljunete na računar pa zatim zalepite na-

Računarska industrija je jedna od retkih privrednih grana kod nas koja nepogrešivo beleži zavidne rezultate. Danas sa ponosom možemo reći da smo po broju fabrika za proizvodnju računara ispred do juče naprikosnovene sile kakva je Amerika. Uplašeni enormnim profitima naše industrije vezane za računare, divovi iz silicijumske doline sve češće bacaju zabrinute poglede na ovo podneblje. Nama, međutim ni to nije dosta i ništa nas ne može zaustaviti na putu ka našem idealu — računar po glavi stanovnika, u svakoj mesnoj zajednici fabrike računara!

lepnicu na njega i na kraju ga opet prodate.

Pri svemu ovome morate voditi računa jedino o krajnjoj ceni vašeg proizvoda. Ona se određuje otprilike ovako: preračunate po dolarskom kursu nabavnu cenu računara u dinare. Cenu nalepnice možete galantno izostaviti. Kod ovog računa, naravno, uzimate najcrnji mogući kurs dolara, od kojeg i samom dolaru dođe tolika muka da još više pozeleni. Zatim da biste bili sasvim fer, na tako sračunatu cenu dodate oko 48 posto za carinu i poreske dažbine (one su vama naplaćene po bančinom kursu). Eventualno (ako vam hitno treba novac) dodate još deset posto za nekakve svoje troškove, zatim stavite prst u uvo i cenu pomnožite odoka najmanje sa pet. Ovaj deo kalkulacije nikako ne smete propustiti — ostali koji se bave ovim vidom proizvodnje neće vam oprostiti i isključiće vas iz tima povlašćenih.

Kao što vidite, posao je izuzetno jednostavan. U isto vreme, to je verovatno jedini posao koji na uloženu marku donosi pet. Toliku produktivnost nema ni jedna industrija sveta. Naravno, pri svemu ovome treba voditi računa o obrtu sredstava, ali tu već sigurno znate sve caka. Održavajte redovne veze sa carinom. Jedino od nje zavisi koliko brzo ćete nabavljati repromaterijal a od toga vam opet direktno zavisi obrt. Dobro ćete proći ako uspete da pronađete način za kupovinu deviza na crno. Na ovom tržištu kurs je samo za oko deset posto viši od bančinog, a vi zarađujete do 500 posto! Jeste li ikad čuli za bolji posao?

Mona Liza na računaru

Ako se na kraju odlučite za ovakav posao povežite se što pre sa onima koji ga već rade. Na taj način obezbeđujete ulaznicu u povlašćeno društvo tehnološki obrazovanih, a u isto vreme obezbeđujete se od većine konflikata, koji mogu nastati između vas i onih koji ovim tržištem već vladaju. Možda vam ova cela priča liči na onaj vic da je Jugoslovenu bilo najteže dok nije izmislio Amerikanca, a sada taj isti Amerikanac misli na njega. No, neka vas to ne brine. Dok god se novac sliva u vašu kasu ovo vam ne mora biti važno. To što ste dali deviza za računar uopšte nije bitno. Dinarska protivvrednost obilato umnožena sliva se u vaše fondove. Vi ste u stanju da kupujući devize nabavljate sve savršenije i skuplje modele računara i, naravno, pri tome ostvarujete sve zavidniju materijalnu dobit. To što je pokvareni Zapad u stvari pronašao novi put za izvlačenje novca vas naravno ne brine, jer pri tome i vi zarađujete, a svojim krajnjim finansijskim efektima doprinosite stabilizaciji. Možda vam izgleda da se ostatak novca samo prelio iz jedne kase u drugu, ali utisci su površni i varaju. Ništa nije istina dok se ne dokaže. Uglavnom vama je dobro, a ostali su zahvalni, jer ste im omogućili da dobiju kartu za voz povlašćenih u novo doba konkurencije u proizvodnji.

Možda vas još pomalo brine pitanje stručnjaka neophodnih u ovoj igri. Samo mirno! Sećam se vremena od pre nekih desetak godina. Otišao čovek na dvonedeljni kurs programiranja u Rovinj. Ostao dve nedelje, vratio se sa gomilom papira ispisanog na računaru. Pohvalio se kako ume da natera glupavu mašinu da ispiše nule po dijagonali papira. Poklonio važnim ljudima Mona Lizu nacrtanu računarom i postao važan čovek. On se najbolje razume u računare. Danas je cenjeni direktor jednog računskog centra. Ako nemate neke probleme, upoznaću vas sa njim. „Samo se javite. On zna sva rešenja.

Sve što ste pročitali bila bi čista idila da u celoj stvari ne postoji i jedna na izgled sitna opasnost. Ceo ovaj posao izaziva do sada nepoznatu bolest (nešto po epidemičnosti slično SIDi). U stvari od predugog držanja prsta u uvu, pri razmišljanju o novim cenama proizvoda, postoji opasnost da vas uvo trajno zaboli za sve. Ne dajte se međutim omesti! Samo hrabro, u nove radne pobede!

Radomir Nikolajev

hard diskovi

Iako je količina podataka koje se smeštaju na disketu u poslednjih nekoliko godina gotovo udesetostručena ugrađivanjem dodatne glave, povećanjem broja traka i uvođenjem dvostruke gustine upisa, flopi disk jedinice su 1984. i 1985. godine došle do svog vrhunca: pojavile su se fizičke prepreke koje praktično onemogućavaju povećanje njihovog kapaciteta. Kakve su to fizičke prepreke? Pre svega, flopi disk se okreće „svega" 300 puta u minutu, što znači da se podaci sa osamdeset traka moraju preneti u memoriju najmanje 80/300=0,26 minuta ili oko 15 sekundi. U primenama u kojima se radi sa bazama podataka problemi se značajno uvećavaju: upisno-čitajuća glava treba intenzivno da se pomera između traka, što znači da će intervali u kojima čitav sistem čeka da se sektor koji treba pročitati nađe ispod glave postajati mnogo duži od vremena potrebnog za efektivno učitavanje podataka. Rezultat: neefikasnost.

I to je sporo

Zašto disk ne bi mogao da se okreće brže? Možda vam je poznato da je upisno-čitajuća glava spuštena do površine diskete, što znači da između dve po prirodi različite površine postoji intenzivan fizički kontakt a samim tim i nezanemarljivo trenje koje se strahovito povećava ubrzavanjem diska. Veće trenje bi, jasno, dovelo do bržeg trošenja glave i same diskete, posle neke granične brzine magnetni sloj na disketi (a samim tim i podaci na njoj) bi bio jednostavno ostrugan.

Disketa je, kao što smo pisali, podeljena na četrdeset ili osamdeset traka, pri čemu ovaj broj ne može bitno da se poveća. Problem nije u tome što je teško preciznije pozicionirati glavu diska, već u tome što se disketa ne nalazi uvek na istom mestu: ako dva puta umećemo disketu u drajv, nećemo je **jednako** snažno gurnuti niti ćemo **jednako** energično pritisnuti klapnu. Osim toga, ne može se očekivati da će azimuti glava raznih drajvova biti **apsolutno** jednaki. Zato između traka mora da postoji dovoljan prostor da se glava može pozicionirati na svaku od njih uz potrebnu toleranciju. Ograničen broj traka povlači, jasno, i ograničen kapacitet diskete od oko 1 Mb.

Poslednje od ozbiljnih ograničenja flopi diskova je broj (kilo)bajtova koji mogu da se upišu na jednu traku: podaci se, kao što vam je verovatno poznato, upisuju promenom magnetnog fluksa kroz materijal. Diskete su, bez obzira na zaštićenost kartonskom kutijom i omotačem, izložene vazduhu koji je prepun čestica prašine, mora se nekako obezbediti da pad nekog takvog zrnceta na disketu ne bude poguban po podatke na njoj, a takvo se obezbeđenje nikako ne slaže sa željom da se podaci na trakama pregusto pakuju!



Deset puta bolji: hard diskovi imaju desetak puta veći kapacitet i deset puta kraće vreme pristupa podacima ali im cena nije deset puta veća od cene flopi diskova

Rešenje sva tri problema može da se dobije jedino suštinskom promenom koja se naziva masivni disk. Ukoliko disketu zamenimo čvrstim magnetnim diskom koji se nikada neće vaditi iz drajva, problem preciznog pozicioniranja se rešava već u fabrici kada se uređaj fino šteluje. Ukoliko je disk, zajedno sa upisno-čitajućom glavom, hermetički zatvoren, nikakva spoljna prašina ne može da ugrozi podatke na njemu, pa je kapacitet jedne trake ograničen jedino karakteristikama materijala koji industrija može da proizvede.

Ostao je još problem brzine okretanja diska: njeno povećanje svakako skraćuje radni vek uređaja. Trebalo bi, dakle, držati glavu na deliću milimetra iznad površine diska (pri tom bi jasno trebalo pojačati sva magnetna polja), što bi moglo da se izvede pomoću veoma precizne mehanike. Kako, međutim, da držite glavu na deliću milimetra iznad diska i da je pri tom brzo šetate od trake do trake? Za rešenje ovoga problema koje ne zahteva basnoslovne troškove je zaista trebalo dosta inspiracije: ako se disk okreće dovoljno brzo, vazduh koji se nalazi uz njegovu površinu će se kretati skoro jednako brzo. Ako je upisno-čitajuća glava dovoljno laka i aerodinamički oblikovana, ona će plivati na tom sloju vazduha kao na nekom vazdušnom jastuku! Rastojanje između glave i površine diska danas varira

Do pre jedva godinu dana flopi disk jedinice su bile san miliona vlasnika kućnih računara koji su se svakodnevno patili sa sporim i nepouzdanim kasetofonima. Mnogi od ovakvih snova su danas ostvareni ali, kako filozofi kažu, apetit dolazi sa jelom. Kapacitet od samo jednog megabajta po disketi je premali, okretanje flopija presporo, podacima se pristupa uz mnogobuke . . . sasvim dovoljno da počnemo da sanjamo o masivnom (hard ili vinčester) disku koji, po sve umerenijim cenama, nudi neverovatno povoljnije karakteristike!



Na samo jedan: U kutiji jednog hard diska hermetički je zatvorene nekoliko diskova, ali sistem funkcioniše kao jedinstvena celina

između 2 mikrometra i 300 nanometara, poređenja radi, svaka dlaka na vašoj glavi je debela oko 70 mikrometara!

Kombinacija masivnog nepomičnog diska u hermetički zatvorenoj kutiji iznad koga pliva upisno-čitajuće glava je poznata pod imenom vinčester *(Winchester) disk* Druga vodeća tehnologija u ovoj oblasti *whitney*, daleko je manje zastupljena kod personalnih računara pa joj nećemo posvećivati posebnu pažnju

Usporeni početak . . .

Winchester diskovi starijih generacija su mogli da se prepoznaju po priličnoj buci zvuk koji izaziva disk u pogonu je unekoliko prigušen zvukom ventilatora koji se obavezno ugrađuje u kutiju Ova je buka konstantna dok flopi disk počinje da se okreće tek kada je računar poželeo da mu pristupa, hard disk se okreće neprekidno jer mu je potrebno nekoliko minuta da postigne radnu brzinu od 3000 obrtaja u minuti i još nekoliko minuta da se iz te pune brzine zaustavi

Zbog ovog ubrzavanja i usporavanja procedura uključivanja računara koji su opremljeni hard diskovima ume da bude prilično komplikovana Najpre se uključuje glavni prekidač a onda hard disk jedinice Sledi podože nestrpljivo posmatranje indikatora *Ready* posle čijeg paljenja smemo da uključimo i sam kompjuter Neki proizvođači računara ugrađuju samo jedan prekidač na čitavu opremu (prekidač se nalazi na monitoru, ne pitajte zašto), a zatim uvode potrebna kola za kašnjenje tako da se računar automatski pali tek kada je disk operativan IBM je taj problem rešio softverski model XT „satima" proverava svoju ispravnost po svakom uključivanju!

Činjenica da glava pliva na vazdušnom jastuku je zaslužna za ogromne kapacitete vinčester diskova, ali i predstavlja izvor problema ukoliko u bilo kom trenutku i iz bilo kog razloga glava padne na magnetnu površinu, podaci na njoj će biti izgubljeni a moguće je i trajno oštećenje diska i/ili glava Gotovo isključivi razlog za ovakve katastrofe je prestanak napajanja uređaja Disk se jasno ponekad mora i isključiti ali je tada situacija drugačija kada pritisnemo taster *Ready* sijalčica će se ugasiti i disk će pomeriti glavu nad posebno određen „parking" na koji će, posle dvadesetak sekundi kočenja, lagano i bezbedno sterirati Moderni hard diskovi su unekoliko obezbeđeni od nestanka struje, pošto im energija u kondenzatorima omogućava da čak i bez napajanja dovedu glavu do „parkinga", ali se ipak treba odreći rada sa računarom u trenucima kada se mogu očekivati problemi sa napajanjem A takvi trenuci nisu kod nas baš tako retki!

. . . i prebrzi rad

Pošto ste nekako povezali hard disk sa računarom (potreban vam je i poseban interfejs jer hard diskovi ne mogu da koriste kontrolere za flopija) počnite da uživate u njegovim karakteristikama Danas se standardni vinčester diskovi od 5 25 inča okreću 3536 puta u minutu (uporedite ovo sa brzinom flopija od 300 obrtaja u minutu) pamte 10, 20, 40 ili 80 megabajta podataka (kod flopija ovaj broj retko prelazi 1 Mb), što zahvaljuju gustini pakovanja od 690 traka po inču (najviše 96 tpi kod flopija) i omogućavaju pristup svakom podatku za samo 8 ms (kod dobrih flopija vreme pristupa obično nije kraće od 100 ms) Brzina prenosa podataka je ponekad prevelika za neke osmobitne mikroprocesore pa se u hard diskove ugrađuju veliki baferi (ponekad se čak primenjuje i dvostruko, pa i višestruko baferisanje) što značajno povećava njihovu cenu Alternativno rešenje je zamena programiranog ulaza—izlaza (koji se zasniva na mehanizmima takozvanog *hand-shaking*-a ili daleko češće nemaskiranih prekida) mehanizmima direktnog memorijskog pristupa (DMA) kod koga mikroprocesor nema udela u prenošenju podataka između periferija i memorije Principe direktnog memorijskog pristupa smo objasnili u prikazima „atarija 520 ST" i „amige"

Jasno je da hard disk nema mnogo smisla koristiti kao brzi kasetofon treba imati dobar operativni sistem i još bolje aplikativne programe da bi se iskoristile blagodati ogromne spoljne memorije i ultra brzog pristupa podacima Što se operativnog sistema tiče neophodna je mogućnost kreiranja stabla kataloga Dok se na jednu disketu obično smeštaju samo datoteke jednoga tipa (na primer bejzik programi tekst koji se obrađuje, poslovni softver i slične stvari) na hard disk smeštamo ogroman broj po prirodi različitih datoteka što znači da je njihovo ostavljanje u jednom katalogu nepregledno i veoma rizično (možda nekada snimimo program PROBA i time obrišemo tekst PROBA) Zato je prirodno u osnovnom katalogu formirati samo nekoliko podkataloga (na primer potkataloga TEKST, BASIC, IGRE PODACI I SYSTEM) a zatim u njima grupisati odgovarajuće datoteke Datoteke u bilo kojem potkatalogu se mogu potpuno nezavisno imenovati tako da nema nikakvih prepreka da koristimo ime PROBA u svakom od njih Kod boljih operativnih sistema možemo da pravimo potkataloge u potkatalozima i tako do neke ograničene dubine

Pri radu sa hard diskom se postavlja i problem zaštite podataka Često se dešava da kompjutera sa masivnim diskovima kori-

ali više ljudi (na primer u kancelariji) i da svaki od njih ne želi da mu se ostali namerno ili slučajno mešaju u posao Tada je prirodno kreirati po jedan direktorijum za svakog od korisnika i zaštititi taj direktorijum lozinkom samo onaj ko zna lozinku će moći da čita modifikuje ili briše programe iz odgovarajućeg direktorijuma (ponekad se čak uvode nezavisna lozinka za čitanje, upis i brisanje) Prilikom formatiranja hard diska (da i hard disk se formatira ali uglavnom jednom u životu) uvodi se i posebna lozinka koju će znati samo vlasnik računara i pomoću koje će moći da poništi ostale lozinke i tako u potpunosti kontroliše svoju opremu

Ako uz hard disk ne nabavite i neki poznati operativni sistem kao što je Unix ili MS DOS načete lako pronaći dobar aplikacioni softver! Ukoliko je naime osnovni operativni sistem vašeg računara dobro koncipiran trebalo bi da programi koji rade sa flopijima rade i sa hard diskom ali to još ne znači da ti programi mogu da iskoriste njegove potencijale Ako je na primer autor nekog *data base* sistema predvideo najviše 1024 sloga veličine od po jednog kilobajta (1024 K obično prevazilazi kapacitete flopija), program jednostavno neće moći da iskoristi 20 megabajta hard diska! Osim toga autori programa koji koriste flopije se trude da minimiziraju broj pristupa disku što znači da ne koriste sve potencijale vinčestera Treba dakle postati Unix kompatibilan ali tada možete da očekujete enormne cene softvera O cenama ćemo međutim, govoriti tek kad razrešimo probleme vezane za sigurnost podataka

Kopija od 800 funti

Vinčester diskovi su, u celini gledano veoma pouzdani pa se često zanemaruje mogućnost gubitka podataka na njima Čak i ako zanemarimo kvarove sasvim je moguće da ćete u nekom trenutku greškom nepovratno izgubiti sve datoteke iz nekog direktorijuma Zato je veoma važno da s vremena na vreme prepišete čitav sadržaj vinčester diska na neki drugi medij obično na diskete Za ovo prepisivanje će vam na žalost biti potrebno 10 20 40 ili možda 60 disketa (zavisno od kapaciteta vinčester diska i flopija koji posedujete)! Umetati dvadesetak disketa u drajv i njihovo snimanje je posao koji može da potraje više od sata što znači da se na njega nije lako odlučiti — možda ćete se radije odlučivati da se pouzdate u svoju srećnu zvezdu Ukoliko tako razonujete poslušajte bar jedan savet **obavezno** prebacite sav softver koji je proizvođač isporučio uz hard disk na flopije (ovaj je softver obično veoma skup i teško ga je samostalno nabaviti) a docnije naredbom COPY prepisujte sva svoja programerska remek dela tekstove i podatke na diskete ako nekada budete morali da formatirate hard disk utrošićete dosta vremena na pretraživanje disketa, ali ćete bar sačuvati rezultate svog rada

Na tržištu mogu da se nađu i specijalizovani uređaji za prepisivanje kompletnog sadržaja hard diska na neku vrstu mikrodrajva iako veoma brzi (čitav se backup završi za dvadesetak minuta i to bez vašeg učešća) ovi uređaji imaju jednu veliku manu — koštaju čak i po 800 funti ponekad više od samog hard diska! Moguća je, naravno i kupovina dva hard diska od kojih će jedan



Ilustracija: Miša Marković

pravashodno biti korišćen za rezervne kopije

Pitanje cene do koga smo u ovom prikazu konačno došli je i dalje bolna tačka masivnih diskova Radi se o veoma preciznim i kvalitetnim uređajima koje nije lako izraditi i koje zato treba dobro naplatiti Smatra se osim toga da su masivni diskovi profesionalni uređaji koje kupuju oni koji uz pomoć računara zarađuju novac A takvi uređaji su dobra prilika za nabijanje cena!

Sam hard disk sa pripadajućim drajvom obično košta između 300 i 400 funti za svakih 10 megabajta Potrebno je naravno kupiti i disk kontroler i odgovarajući operativni sistem što će vas koštati daljih 200 funti Ako vas suma od oko 1000 funti za disk od 20 megabajta ne plaši uzmite u obzir i cene softvera kada jednoga dana stupite u profesionalni svet cene engleskih programa koje su vam izgledale astronomske u poređenju sa piratskom ponudom postaju prava pesma Vrlo je verovatno da ćete za dobar tekst procesor bazu podataka i spreadsheet platiti najmanje 1500 funti, fortran kompajleri i slični sistemski programi mogu da budu i daleko skuplji! Zbog toga sa žaljenjem možemo da završimo ovaj napis konstatacijom da će iako cene hard diskova neprekidno padaju korišćenje njihovih potencijala zadugo izmicati mogućnostima jugoslovenskog hakera

Dejan Ristanović

PROIZVOĐAČI I MODELI

Ukoliko vas je naš napis zainteresovao toliko da ste počeli da razmišljate o hard disku zanimaće vas podaci o mogućnostima povezivanja popularnih personalnih modela sa njima Podimo (abecednim) redom

Apricot F1 *je najprirodnije povezati sa „oficijelnim" modelom vinčester diska MSD 3 5 Za 1100 funti ćete dobiti dva praktično nezavisna uređaja od kojih je jedan običan izvor za napajanje a drugi disk od 10 M! Operativni sistem se odlično uklapa u MS DOS tako da hard disk biva automatski konfigurisan kao drajv A Vaš stari flopi postaje drajv B*

Atari 520 ST, *sa svojim izvanrednim karakteristikama prosto plače za hard diskom Na žalost iako je na nedavnom sajmu časopisa PCW prikazan masivni disk povezan sa modelom 520 ST, Atari za sada ne pominje izlazak na tržište specifikacije niti eventualne cene*

BBC *mikrokompjuter nije teško povezati sa hard diskom — za 1300 funti Acorn prodaje vinchester disk od 10, a za 1900 funti disk od 30 megabajta U cenu je uračunat i interfejs (File Server Level III) kao i ROM sa ADFS-om operativnim sistemom koji omogućava kreiranje hijerarhijskog stabla kataloga zaštitu direktorijuma i mnoge druge stvari Za daleko manje novca (460 funti) se od Solidiska (0702 354674) može nabaviti hard disk od 10 megabajta sa pripadajućim softverom koji je u mnogim aspektima bolji od Acornovog Na žalost kompatibilnost ova dva sistema je prilično sumnjiva Može se najzad kupiti i torchov Unicorn koji zajedno sa mikroprocesorima MC68000 i Z80 nudi 512 K RAM-a 20 M hard disk i čuvene operativne sisteme Unix i CP/M Cena je „samo" 2450 funti*

Sinclair QL *se za razliku od prethodnih računara može opremiti jedino hard diskovima nezavisnih proizvođača Firma CST (zovite ih na 0223 323302) prodaje hard disk od 10 M zajedno sa potrebnim interfejsom za 1200 funti Za dodatnih 400 funti možete da dobijete i jedan flopi disk pošto ćete se teško odlučivati da radite backup na preko 80 kaseta za mikrodrajvove Karakteristike uređaja su veoma povoljne posebno vreme prenosa od 1.6 mikrosekundi po bajtu*

Sanyo 550 i 555 *možete da snabdete masivnim diskovima firme Logitek (0257 426644) Cene su na žalost previsoke 2000 funti za 10 M i oko 1200 za svakih sledećih deset Mogu se za uzvrat nabaviti i „džinovski" diskovi od 80 M*

Peek & poke show

Reklama svuda oko nas

Reklamiranje priloga o računarima počelo je i pre pojave računarskih listova što je i logično kada se ima u vidu već poslovična nesposobnost naših struktura što neformalnih što formalnih da izađu u susret narodnim željama u svakom pa i računarskom pogledu Dakle prvo su u drugim listovima nastali prilozi o računarima pa specijalizovani listovi o računarima a između te dve faze u našoj računarskoj revoluciji došlo je do podfaze reklamiranja pomenutih listova a koju takođe možemo podeliti na tri oblika i to reklamu u novinama na radiju i na televiziji

Štampa

Pošto su novine najstarije najklasičnije i najnepromenljivije sredstvo javnog informisanja jasno je da i reklama u novinama ne može da bude mnogo drugačija nego reklama za druge neračunarske listove pa se stoga dešava da se na isti način reklamiraju, na primer „Računari" „NIN" „Danas", ali i „Praktična žena" i (o, zar se i to može?!) „Tempo" i „Zvezdina revija" Međutim u novinama se malo može uraditi pa da pogledamo šta je s

Radijom

gde je situacija nešto bolja, jer se računarski listovi ne reklamiraju onako kako se reklamira keks Đetić ili, pak nove ploče Hamida Ragipovića Beska, nego se čak koriste i programčići pomoću kojih se može očitati kompletan sadržaj lista To što je taj zvuk prilično neprijatan čak razdražujući za slušaoce koji vole već pomenutog Beska ili „Poslednju igru leptira" nije za nas bitno jer je to vrsta ljudi kojoj poput mamuta i australopitekusa preti izumiranje A mi kojima pripada budućnost, krećemo korak dalje, tj u

Televiziju

koju ovog puta treba pohvaliti jer je predstavljanje računarskih listova i časopisa urađeno na prilično moderan način korišćenjem mogućnosti medija mada ne u potpunosti, jer recimo nikad nisam video da se neki list („Računari", na primer) oglašava pomoću neke šaljive, duhovite, da ne kažem lepršave kompjuterske igre recimo popularnog Pećinara (vlasnici komodora koji vole da se igraju znaju na šta mislim) No, reklamama (na televiziji i uopšte) nećemo biti zadovoljni sve dok se budu radile u paketima sa drugim robama i dok ih budu radili ljudi kojima je do računara stalo toliko koliko im reklama za iste donosi bodove varijabile što može poslužiti i kao

Zaključak

PEEK & POKE SHOW
Raspisuje javni nagradni konkurs za

„NAJGORI MIKRORAČUNAR U 1985."

Ako želite da učestvujete u ovom konkursu potrebno je samo da nam pošaljete razglednicu dopisnicu ili pismo (može i telegram) sa vašim mišljenjem o tome koji računar je bio najgori u upravo proteklој godini Ako imate komentar ili objašnjenje zašto je upravo vaš izbor tej za koji mislite da jeste napišite ga to će nam puno pomoći

Nagrade

Kada izbrojimo sve negativne glasove i kada ustanovimo koji je računar zvanično izabran za najgori u 1985 onda ćemo iz gomile onih koji su glasali za njega uz pomoć našeg specijalnog gosta (neka njegov identitet za sada ostane tajna) izvući tri srećnika koji će dobiti nagrade koje uzbuđuju

1 nagrada — kaseta na kojoj kompletna redakcija „Računara" peva pesmu „Program tvog kompjutera"

2 nagrada — kaseta na kojoj urednici „Računara" pevaju pesmu „Soba 23"

3 nagrada — kaseta na kojoj priređivači „Peek & poke show-a" pevaju pesmu „Je sam lažljiva"

Rezultati

Rezultati konkursa i imena nagrađenih će biti objavljeni u „Računarima 13"

Ako odziv na konkurs bude veliki nabavićemo još nekoliko specijalnih nagrada i dobiće iznenađenje

Pišite na adresu „Peek & poke show" Reč — Galaksija BIGZ Bulevar vojvode Mišića 17 Beograd

Vašar oplemenjuje

Izgleda da su računari stekli pravo građanstva (ili seljanstva) koje im kod nas tako dugo tali Da oni su postojali i pre ali nisu baš pripadali ovom podneblju nešto kao banane ili papagaji Srećni događaj ponarodnjavanja računara dogodio se na karmasu nazvanom Novogodišnji vašar Pažljivi posmatrač i posetilac je mogao da primeti da pored ćevapčića originál poliesterska svila predratna moda ogledalcadi i sličnog probranog kiča na Vašaru postoji i promocija Komodorovog računara C-128 koji je organizovao Sin-Com klub iz Beograda Tako su najzad računari spušteni sa svog ezoteričnog filmskog mesta u narod i šira To su verovatno prvi udari računarske revolucije koja samo što nije počela Sad će samo što nije Zato bi trebalo pažljivo zapisati datum kada su se računari pojavili na Velikom Vašaru Možda ćemo to slaviti jednog dana

Uticaj kurseva na svetsko neznanje

Danas više ne možete da otvorite novine a da ne naletite na pravu gomilu reklama i oglasa za male škole bejzika ubrzane kurseve mašinskog jezika i slična rasprodaja znanja Ideja nije naročito originalna ali je — za organizatore — izuzetno korisna To je evoluirana šema razvijena još pre mnogo godina na kursevima fotografisanja dopisnim školama šivenja slepim kursevima slepe daktilografije i sličnim pametnim tezgama Ovog puta tema je kako živeti od hakera? izgleda da nije teško „Peek & poke show" je veoma diskretan pa zato neće pominjati imena ali trenutno u našem glavnom gradu postoji desetak (najmanje) „kurseva" koji pružaju onoliko računarskog znanja koliko i Mikijev Almanah To i nije tako malo kad se uzme u obzir kakvi su neki predavači Da bi obrazovanje (i zarada) bili kontinualni organizatori su se potrudili da razbiju kurseve u male malecke celine pa tako imamo početne srednje, više usavršene specijalne ubrzane i kondenzovane kurseve Pravo bogatstvo za nas! Dobro gledajte gde čete da naučite vaš bejzik Raspitajte se malo! Grad je pun ljudi koji su svoj kurs blistavo završili i sad se blistavo kaju Kursevi suzama ne veruju

Pripremaju
Branko Đaković i Saša D. Kovačević

Dejanove pitalice

Nije lako biti (de)šifrant

Drugi nagradni zadatak je očigledno bio prilično tvrd orah za naše čitaoce — u predviđenom smo roku dobili ukupno 12 potpuno tačnih, 26 umereno tačnih i dva nepotpuna odgovora

Najjednostavniji način da se pride rešavanju problema je pretpostaviti da se na kraju teksta, kao što priliči, nalazi tačka, što znači da je broj '1' zamena za tačku. Sada možemo pretražiti čitav tekst i pronaći sve jedinice, iza svih njih se nalazi znak za stepenovanje (^) koji, dakle, predstavlja zamenu za blanko. Zatim pomoću računara treba prebrojati slova u tekstu i ustanoviti da se najčešće javlja malo h koje je, dakle, verovatno zamena za slovo a. Sledeću nešto nižu frekvenciju ima zarez koji je, dakle, verovatno oznaka za slovo e. Na sličan se način da pretpostaviti da je slovo u zamena za slovo i. Zatim treba napisati tekst i pogađati neke reči i konstrukcije, ne zaboravljajući da slovo koje identifikujete treba zameniti kroz čitav tekst.

Tačno rešenje zadatka je dato na slici. Do njega se, međutim, nije moglo doći potpuno egzaktno: vlastita imena koja započinju velikim slovima, kao što je Robin Brodhed, Esi, Triša ili Albert Ajnštajn, se mogu identifikovati jedino po sluhu ili čitanjem knjige Kapija 2 od Frederika Pola. Pokazalo se da je relativno mali broj rešavača pročitao ovu knjigu (prava šteta!) pa smo se načitali svakakvih imena (kako bi se Robin Brodhed osećao da je čuo da mu se žena zove Psi?). Zato smo priznali sva odgovore sa greškama u velikim slovima, ali ne i odgovore koji su zanemarili zagrade u 14 i 16 redu — one su se mogle egzaktno identifikovati. Tako smo došli do spiska od 12 potpuno tačnih rešenja koja su poslali Milan Blagojević, Petar Veselinović, Vladan Dugarić, Esad Milaković i Bratislav Ivenčić iz Beograda, Stevan Hunc iz Zagreba, Jero Lajović iz Ljubljane, Matej Balčić iz Maribora, Predrag Valente iz Niša, Imre Vencel iz Zrenjanina, Jovan Petrović iz Siska i nepotpisani čitalac iz Niša. Najviše sreće je imao **Predrag Valante** koji je izvukao I nagradu od 10 000 dinara. Treba, međutim, da kažemo da smo bili u velikom iskušenju da dodelimo prvu nagradu Slobodanu Jankoviću iz Niša koji je na neki volšeban način uspeo da pogodi seed koji je korišćen za generisanja šifra. Mislili smo ipak da nema smisla favorizovati vlasnike pojedinih računara, a pronašli smo i par greščica u rešenju druga Jankovića.

Posle izvlačenja prve nagrade pomešali smo preostala potpuno tačna rešenja sa pismima koja su sadržala odgovore ocenjena kao skoro tačni (na primer greška u zagradama) i izvukli drugu i treću nagradu koje su dobili Dragan Pujić iz Beograda i Matej Balčić iz Maribora.

Naš novi nagradni zadatak je vezan za takozvane spiskove koji su u poslednje vreme neobično popularni u Beogradu. Da objasnimo najpre u čemu je stvar: neko vam za 2000 dinara proda spisak na kome se nalazi 12 imena i adresa. Vaša je obaveza da prvom čoveku sa spiska uplatite 2000 dinara poštanskom uputnicom i da zatim uklonite njegovo ime sa spiska a svoje podatke upišete na njegov kraj, na dvanaesto mesto. Zatim pronađite dva prijatelja kojima prodate spisak za po 2000 dinara, čime ste u potpunosti „izvadili štetu". Ostaje vam samo da čekate: kroz desetak dana će na vašu adresu pristići otprilike jedna stara milijarda! Vaše ime je naime prosperiralo prema čelu spiskova koji su se umnožavali poput lavina, pa vam je ogroman broj nepoznatih ljudi uplatio novac!

Zašto se na ovaj način svi ne bismo obogatili? Problem je u tome što se spiskovi strahovito brzo množe, što znači da posle određenog broja „generacija" nesrećnici koji kupe spisak neće imati kome da ga prodaju, tako da će u stvari ogroman broj ljudi biti u malom gubitku da bi malobrojni bili u ogromnom dobitku (u ogromnom dobitku je jasno i pošta, ali to nije predmet ovoga razmatranja).

Pretpostavimo da Jugoslavija ima tačno 22 miliona stanovnika koji su platežno sposobni i da se igra ne širi izvan naših granica. Pretpostavimo dalje, da je svako kome ponudite spisak **dužan** da ga kupi, osim u slučaju kada je već učestvovao u igri. Pretpostavimo dalje da se osoba koja još nije učestvovala u igri može brzo i lako locirati bez obzira koliko daleko živela. Igra započinje i traje dok ima ljudi koji su dužni da kupuju spiskove. Koliko će Jugoslovena biti u maksimalnom a koliko u bilo kakvom dobitku?

Rešenja zadataka, zajedno sa kuponima, treba poslati na adresu: Galeksija (Za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd, tako da pristignu u redakciju pre 20. februara 1986. Imena nagrađenih (10 000, 5 000 i 3 000 dinara) će biti objavljene u „Računarima 13".

```
Qh^H,^3$',G^*$@už^&y$'J,'^u
Ja se zovem Robin Brodhed i

žhX@$phWuXh^HhG^$H$@h^b^"uWh'$G
najbogatija sam osoba u čitavom

Hbž',''$G^HuHW,Gbl^Q,'užh^$H$@h
sunčevom sistemu. Jedina osoba

$$Xh^Gu^X,^b^W$G^_$p<,'b^_yu@<usžh
koja mi je u tom pogledu približna

X,HW,^HWhyu^&$',yw^h^$ž^@u^Gu^@u$
jeste stari Bover, a on bi mi bio

X$E^Gž$p$^_yu@<usžuXu^'h^_$<$'užb
još mnogo približniji da polovinu

H'$ph^ž$'$h^žuX,^@h$u$^žh
svoga novca nije bacio na

b$<hžXhžX,^b's,yu$h^u^$@žh'<XhžX,
uklanjanje udžerica i obnavljanje

pyh'H$uJ^žhH,<Xh^u^'h^$'
gradskih naselja i da od

_y,$HWh<uJ^_hyh^'$@hy^',$^žuX,
preostalih para dobar deo nije

bWy$Eu$^žh^_y,Wyhsu'hžX,^H'h$,
utrošio na pretraživanje svake

HW$_,^H',GuyH$$p^_y$HWyhžHW'h^b
stope svemirskog prostranstva u

_$Wyh3u^3h^@y$'$G^Hh^$HWh$uGh
potrazi za brodom sa ostacima

žX,p$',^s,ž,w^ZyuE,^i Wh^(,^Hh
njegove žene, Triše (šta će sa

žX$G^'h^byh'u^h$$^X,^@b',
njom da uradi ako je bude

_y$žhEh$w^Xh^ž,GhG^_$XGh1Ɛ1
pronašao, ja nemam pojma.).

@'yh'<X,^O$X,^s,ž,^CHu^X,^ž,^G$s,
Zdravlje moje žene Esi je ne može

@uWu^@$<X,1^Qh^X,"'$<uG1^(h'
biti bolje. Ja je volim. Kad

bGy,Ow^EW$^(,^y,žU^$h'^žu
umrem, šta će reći kad mi

G,'u$užH$u^$$O_<,W^'uE,^ž,^@b',^b
medicinski komplet više ne bude u

HWhžXb^'h^G,^3h$y_uw^uGhG^X,'hž
stanju da me zakrpi, imam jedan

Gh<u^_<hž^$h$$^'h^H,^_$HWb_u^Hh
mali plan kako da se postupi sa

X,'ž$G^'ybp$G^$H$@$G^$$Xb^'$<uG
jednom drugom osobom koju volim

u^W$^Gu^_ybsh^$H,$hX^3h'$'$<XHW'hi
i to mi pruža osećaj zadovoljstva.

6$W$'$^Gu^H',^_ybsh^$H,$hX
Gotovo mi sve pruža osećaj

3h'$'$<XHW'hi^Q,'užu^u3b3,Wh$^X,
zadovoljstva. Jedini izuzetak je

G$X^Hh',W$'h'h$^3h^žhb"žh^_uWhžXhw
moj savetodavac za naučna pitanja,

$$G_XbW,yH$u^_y$pyhG^1<@,yW
kompjuterski program Albert

1XžEWhXžw^$$Xu^ž,_y,HWhž$^žhHW$Xu
Ajnštajn, koji neprestano nastoji

'h^Gu^$@XhHžu^zhJ$'^_yuž$u_1
da mi objasni Mahov princip.
```

Dejan Ristanović

_______________ Jugoslovena će biti u dobitku, od toga
_______________ u maksimalnom.

Ime i prezime _______________

Adresa _______________

Mesto _______________

Kompjuterske mozgalice

Neupisivo upisivanje

Digitalni optički diskovi predstavljaju novi memorijski medij koji pruža nove perspektive. Disk prečnika 12 inča (1 inč=2,54 cm), sa cenom manjom od 100$, može da memoriše više od 100 milijardi bita pri današnjoj tehnologiji, a u neposrednoj budućnosti i znatno veće količine podataka pri još boljem odnosu veličina memorije/cena. Ovolika količina podataka zahteva, na primer, oko 40 koluta magnetne trake, ali je vreme pristupa podacima na trakama daleko veće od 1/10 sekunde, koliko je potrebno kod video diskova

Digitalni optički diskovi, međutim, imaju i jednu neugodnu osobinu za većinu korisnika. Informacije se na njih upisuju samo jednom! U računarstvu, međutim, ništa nije nemoguće

Tanak sloj telurijuma, na kojem se pomoću lasera stvara submikronska rupica čime se označava prelaz sa inicijalnog stanja „0" u stanje „1" ako se želi upis jedinice, ne može ničim da se regeneriše. Pri čitanju laser osvetljava svaku poziciju i smanjena refleksija rupice lako se registruje. Veliki potencijal novog memorijskog medijuma naveli su Ronalda Rajvesta (Ronald Rivest) i Adi Samira (Adi Shamira) a zatim i veći broj drugih istraživača da preispitaju suštinu jednoupisivosti optičkih diskova. Oni su uveli i posebno ime za ovu vrstu memorija „woms" (write-once memories)

Pored digitalnih optičkih diskova u jednoupisive memorije, spadaju i PROMS (programmable read-only memories) kod kojih se topi spoj koji smo selektivno izabrali a omogućavao je protok struje, papirne traka i papirne kartice. Ako imate neki od ovih memorijskih medijuma pokušajte da ih više puta iskoristite.

Ideja o višeupisivosti jednoupisivih memorija zasniva se na rezultatima teorije kodovanja i može da se objasni na sledećem jednostavnom primeru. Pretpostavimo da smo željenu informaciju zapamtili kodovanu brojevima 00 01 10 i 11 tako što smo koristili inicijalnu kolonu tablice 1 zapisa. Ako u sledećem upisivanju želimo da ostane upisan isti broj, onda ništa ne menjamo u zapisu, a ako želimo da upišemo neki drugi broj upisujemo sa kodovanog kao u novoj koloni tablice 1

*Dekodovanje zapisa je jednostavno reč abc predstavlja 2-bitnu vrednost (b*c) (a*c) bez obzira u kojoj se koloni nalazi zapis tako da pri novom upisivanju u memoriju moramo prvo da iščitavamo memoriju, ali pri čitanju memorije ne vodimo računa o tome da li je memorija upisivana jednom ili dvaput. Znakom * je označena opreacija „ekskluzivno ili" koja je realizovana na čipu. Rejvest i Samir su dali odgovor i na pitanje u vezi sa graničnim problemom za koliko je manji broj bita koje možemo da upišemo pri N-tostrukom upisivanju nego pri samo jednom upisivanju. Deo njihovog odgovora dat je u tabeli 2. U prvoj koloni je dat broj upisivanja a u drugoj veličina smanjenje memorije pri takvom upisivanju.*

Pre nego što pogledamo pregled nekih kodova za višekratno upisivanje na jednoupisivu memoriju primetimo da su digitalni optički diskovi posebno pogodni za čuvanje podataka koji su manje-više statični, kao na primer programi, dokumenti, slike ili baze podataka kod kojih se ažuriranje podataka ne vrši odveć često, a takav deo baza podataka je velik. Posebnim tehnikama organizacije podataka na digitalnim optičkim diskovima mogu se potpuno eliminisati potrebe za magnetnim diskovima. Profesor David Klarner (koji predaje matematiku na Univerzitetu države Njujork u Bingamptonu) generisao je elegantan ciklični kod inspirisan klasičnom teorijom kodovanja koji omogućava trostruko upisivanje. Kod je dat u tablici 3

Koristeći Klarnerovu tehniku student Nebraska univerziteta u Linkolnu David Livi je generisao kod koji omogućava četiri upisivanja koristeći 7 bita. Među ostalim autorima kodova ističe se Džems Sakse sa Karnegi Melon univerziteta u Pitsburgu koji je pri tom koristio rekurziju i svojstva simetrije. Ako vam se svideo princip višeupisivosti na jednoupisive memorije a želite da konstruišete nov kod požurite — u literaturi a potako i u praksi stalno ih je sve više. Ipak, pored konstrukcije novih kodova ostavljeno je još mnogo drugih pitanja koja traže odgovor (na primer da li je svaki novoupisan kod veći numerički od prethodnog ako se računa zapis kao broj zapisa u binarnom broju)

- *koliko je kompleksnost dekodovanja i preupisivanja*
- *kako prevazići eventualne greške pri upisu,*
- *da li se svojstva višeupisivosti mogu primeniti i u drugim oblastima računarstva i nauka*
- *kako upisivati u memorije koje imaju tri ili više mogućih stanja tako da se ova optimalno koriste i*
- *koji su kodovi praktično korisni*

Miodrag Potkonjak

tablica 1

%	inicijalno stanje	novo stanje
00	000	111
01	100	011
10	010	101
11	001	011

tablica 2

broj upisivanja	stanje pri jednom upisivanju
1	1 000
2	1 294
3	1 549
4	1 783
5	2 003
10	2 983
20	4 668
50	8 960
100	15 191
200	26 346

tablica 3

%	prva generacija		druga generacija		treće generacija
0	10000		01001		01111
		ili	00110	ili	10110
				ili	11001
1	01000		10100		10111
		ili	00011	ili	01011
				ili	11100
2	00100		01010		11011
		ili	10001	ili	10101
				ili	01110
3	00010		00101		11101
		ili	11000	ili	11010
				ili	00111
4	00001		10010		11110
		ili	01100	ili	01101
				ili	10011

Naš test

video digitajzer

U prošlom broju „Računara" pisali smo o mogućnostima digitalizacije zvuka na kućnom računaru. Ovoga puta pripremili smo test uređaja za digitalizaciju slike — video digitajzera firme Print-Tehnik iz Minhena. Uređaj koji smo mi testirali radi na „komodoru 64", ali se iste ili slična sprava mogu nabaviti i za ostala računare. Iako se još uvek nalaze u primitivnoj fazi, ovakvi uređaji značajno proširuju upotrebnu vrednost računara.

Video digitajzer je uređaj koji digitalizuje bilo koju sliku i prebacuje je na ekran i u memoriju računara za dalje korišćenje. Do pre nekoliko meseci ovakvi uređaji su izgledali kao pravo čudo tehnike. Bila je to velika zastrašujuća ploča sa ogromnom količinom delova. Napretkom tehnologije uređaj je sveden na dimenzije plastične kutije dugačke 8, široke 6 i debele 2 centimetra koja se odlično uklapa u pozadinu računara. Na samom uređaju se nalaze priključak za video kameru i dva mala potenciometra za podešavanje osvetljenja i kontrasta. Sam uređaj se priključuje na paralelni izlaz računara i radi vrlo jednostavno. Uz uređaj se dobija i jedna disketa, na kojoj se nalaze programi koji mu omogućavaju da radi i nekoliko demonstracionih programa.

Kada učitate program, na ekranu dobijete lepo urađen meni. Pored digitalizacije, autori softvera su najviše pažnje posvetili štampanju i snimanju slike. Kursor treba dovesti do prozora na kojem piše DIGITIZE i pritisnuti RETURN. Ekran će se zamračiti i proces digitalizacije će početi. Ceo proces digitalizacije jedne slike traje oko četiri sekunde i posle njega na ekranu se dobija deo slike koju ste digitalizovali. Digitajzer nije ograničen samo na rezoluciju samog računara. Kod „komodora" uređaj digitalizuje sliku veličine 480 puta 400 tačaka, pri čemu se samo deo vidi na ekranu. Pomoću tastera za vođenje kursora ekran se može šetati po slici sve dok se ne odabere najbolji položaj. Ovo je mnogo zgodnije jer kameru možete fiksirati a sliku pomerati u svim pravcima.

Obrada slika

Kada dobijete prvu sliku, nemojte se odmah razočarati (kao mi) ako slika bude loša. Potrebno je malo znanja i prakse da bi se dobile najbolje slike. Pri tom je vrlo važno podesiti osvetljenje i kontrast na digitajzeru, jer to bitno utiče na kvalitet slike.

Kada dobijemo onakvu sliku kakvu smo želeli, vraćamo se u meni i biramo opciju koja nam najviše odgovara. Kao prvo, samu sliku treba snimiti i sačuvati za buduću upotrebu. To možemo uraditi na više načina. Moguće je pre svega snimiti kompletnu sliku koju smo dobili (480×400), ali ćemo takvu sliku moći da koristimo samo uz softver samog digitajzera. Na sreću, na raspolaganju imamo još nekoliko mogućih



Korak dalje: Video digitajzeri značajno proširuju upotrebnu vrednost računara

Snimio: Vladimir Simović

načina snimanja. Najzgodnije je sliku snimiti tako da se posle može ponovo iskoristiti u nekom programu za crtanje ili grafiku. Na raspolaganju su vam KOALA-SAVE i PAINT-SAVE komande pomoću kojih se dobijaju fajlovi na disku koji se mogu učitati u KOALA-PAINTER ili PAINT-MAGIC. Nakon toga na slici se mogu vršiti sve manipulacije koje dozvoljavaju ovi programi. Ovi fajlovi se lako mogu preraditi i za ostale programe za crtanje (DOODLE, BLAZING PADDLE), a korisno je i sliku prebaciti u oblik kakav će moći da se učita u SIMON'S Bejzik ili neki drugi prošireni bejzik (G GR). Tako možete slike koje ste dobili i vrlo praktično da primenite.

Pošto smo sliku sačuvali za dalju upotrebu, mogli bismo i da je odštampamo. Nijedan vlasnik „komodora" ne treba da se plaši da program neće raditi sa njegovim printerom. Na raspolaganju su vam opcije za sve printere koji se mogu priključiti na „komodor". Možete, čak, koristiti i paralelni printer — i on je predviđen programom. Treba samo da kažete koji printer imate i kako je priključen i štampanje će početi. Program štampa kompletnu sliku, znači 480×400, pa je moguće dobiti dosta veliku sliku na papiru. Na disketi je dat i poseban program koji na Epsonovim i još nekim kvalitetnijim printerima štampa slike u 16 crno-belih nijansi, što izgleda zaista impresivno. Ukoliko želite sliku manjeg ili još većeg formata, možete je odštampati i iz nekog drugog programa koji podržava štampač.

Savršeni alarm

Pored programa za šesnaest boja, na disketi se nalazi još nekoliko interesantnih i korisnih programa. Jedan od njih je i alarm. Fiksirate kameru u jednom pravcu, digitalizujete sliku i uključite alarm. Računar će svakih pet sekundi ponovo digitalizovati sliku i u slučaju da se pojavi neka razlika između dve slike, začuće se grozan zvuk sirene za uzbunu koja će raditi sve dok ne pritisnete neki taster na računaru, a tada sve ide iz početka. Pre startovanja alarma možete odrediti računaru na koliku promenu da reaguje. Program, u stvari, radi tako što sabira binarne vrednosti cele slike i ako razlika zbirova dve slike pređe određenu vrednost, uključuje se zvuk sirene. Osim ovog programa, na disketi se nalazi još nekoliko korisnih programa i nekoliko demonstracija. Posebno je interesantna digitalizovana slika prodavca iz radnje koja uređaj prodaje, jer je slika zaista urađena na zavidnom nivou.

Sve u svemu, jedan vrlo interesantan dodatak za vaš računar, koji bi mogao da ima vrlo široku primenu. Može se koristiti za izradu slike u finoj grafici, kao čuvar (alarm) ili za nešto sasvim treće. Ako imate veliku količinu raznih slika ili crteža koje treba čuvati, možete da ih sve digitalizujete i spremite na nekoliko disketa.

Vladimir Krstonošić

SUPERKOMPJUTERI '86.

džinovi u boci

Rešili ste da kupite novi računar? U redu, kakav bi on trebalo da bude? Pa, naravno, što brži, sa što većom memorijom i . . . Ako imate desetak miliona dolara i strpljenja da sačekate kraj (ili bar sredinu) ove godine, a zadovoljavaju vas brzina od nekoliko hiljada megaflopova (flop je jedna operacija u pomičnom zarezu u sekundi, a megaflop je milion flopova — IBM 3081 K operiše sa oko 2,3 megaflopova, VAX 11/780 sa oko 0,3, a VAX 11/750 oko 0,2 megaflopa) i memorija od nekoliko milijardi bajtova, onda pažljivo pročitajte tekst, proučite tabele i odaberite nešto dobro za sebe ili svojoj devojci za rođendan.

Što treba gledati kad kupujete superkompjuter? Isto što i kod bilo kog drugog računara: hardver, sistemski i aplikativni softver.

Egzotične tehnologije

Hardver je osnovni preduslov za brži računar. Veće brzine postižu se, zahvaljujući pre svega povećanoj brzini komponenti, smanjenju vremena prenošenja informacija smanjenjem dužine veza između komponenti i povećanjem gustine pakovanja ili promenama u arhitekturi koje omogućavaju paralelno izvršavanje operacija. Povećanje brzine komponenti je ograničeno u najoptimističkijim slučajevima, brzinom svetlosti ili nekom drugom brzinom koju daje teorijska fizika. Tri tehnologije koje najviše obećavaju u ovoj i nekoliko sledećih godina su brza CMOS kola koja se hlade tečnim nitrogenom sa temperaturom od 77 Kelvinovih stepeni, gusta bipolarna integralna kola i GaAs kola koja će, najverovatnije dominirati i početkom sledeće dekade.

Krajem veka i početkom sledećeg doći će do izražaja i naprednije i egzotičnije tehnologije koje su još u fazi ispitivanja: molekularne epiksijalne komplementirani galijum-arsenid, galijum-aluminijum arsenski tranzistori sa velikom mobilnošću elektrona i Džozefsonov (Josephson) spoj. Najviše nade se polaže u GaAs-AlGaAs tranzistor koji može da menja stanje za vreme koje se meri pikosekundama (milionitim delovima milionitog dela sekunde) a istovremeno zrače izuzetno malo toplote. Sve ostale tehnologije su bar za red veličine slabije od ove. Ovaj tranzistor se naziva i HEMT (high-electron-mobility tranzistor) ili MODFET (modulation doped field-effect transistor) ili TEGFET (two-dimensional electron-gas FET) ili SDHT (selectively doped heterojunction transistor) ili

U znaku Unixa

Sistemski softver doprinosi povećanju brzine generisanjem jezika koji omogućavaju efikasnost u kodovanju onih algoritama koji je u skladu sa arhitekturom. Do ove godine operativni sistemi su bili najbolnije tačka superkompjutera. Primera radi, Sajber 205 nema mogućnost istovremenog rada više korisnika, što znači da nije zgodno ulaziti u kupovinu sa prijateljima. Nedav-



Najbrži na svetu! Novi „krej", „krej-2", sastoji se od 240 000 čipova koji troše između 300 i 500W, ali mu to ne smeta da radi sa klokom od svega 4,1 nanosekunde (procesor Z80A je sporiji 60 puta)

no je Convex Computer Corp iz Ričardsone u Teksasu implementirala na svom 64-bitnom superračunaru Convex C-1 operativni sistem Unix 4.2. Po rečima Bertona Smita, šefa Denelcor inc. iz Aurore u Koloradu za njihov HEPe HEPi Krajova kompjutera Unix je rešenje za problem oko operativnih sistema, iako ima i drugačijih mišljenja.

Glavni programski jezik većine superkompjutera je fortran. Fortran kompajler imaju STAR-100, svi Krajovi (Cray-1 dva: CFT i CIVIC, Cray X-MP ima takođe dva: CFT 110 i CFT 113+), 100-megaflopni ASC firme Texas Instruments i Burroughs-ov BSP. Sajber 205 ima kompajler za fortran 4 a i CDC-ov metasembler. Na HEP-ovima je realizovan fortran koji ima posebne pozive (Create) koji omogućavaju odlično korišćenje arhitekture. Britanski tridesetdvobitni matrični procesor DAP (Distributed Array Processor) firme ICL koji može da izvrši 125 miliona sabiranja i 11 miliona množenja u sekundi ima, pored fortran kompajlera i algol kompajler a i poseban jezik DAP. NASA je kupila za obradu slika Zemljine površine sa satelita kod američke firme Goodyear Aerospace Corp za 4,7 miliona dolara izuzetno moćan matrični procesor MPP (Massively Parallel Processor) koji je izrađen integralnim kolima sa silicijumskim strukturama na safiru, a ima 6 milijardi sabiranja u sekundi i 2 milijarde množenja kada je dužina reči 6 bita a dve odnosno devet puta manju brzinu kada je reč dužine 12 odnosno 32 bita. Ovaj procesor se međutim ne može programirati ni na jednom jeziku visokog nivoa. Najjači matrični procesor Burroughs-ov Phoenix koji može da uradi više od 10 milijardi operacija nad 64 bitnim rečima a koji preporučujemo ljubiteljima dobrog zvuka jer se koristi za rešavanje akustičkih zadataka programira se takođe na fortranu.

Bejzik i paskal ne postoje ni na jednom superkompjuteru ali zato za one kojima je bliska ada a imaju veze u američkoj armiji preporučujemo procesor Mark IIe koji je proizveden u Lourensovoj laboratoriji u

Livermoru On radi na adi brzinom od 400 megaflopova dužinu reči od 18,36 i 72 bite a košta svega 300 000 dolera Koga ove brzine ne zadovoljeve, e ne može bez programiranje ne edi neka za 6 milione dolera kupi, neravno, opet eko ima jeke veze S-1 Merk IIA brzine 1000 megaflopova Ineče, ovej procesor koristi emerička momerica za brzu obradu podetaka i upravljanje gadenjem, pe je izuzetno pouzdan i ima poseban procesor za dijegnostiku i remont Niska cena je postignuta korišćenjem eutometizovenog eletem projektovanja SCALD (Structural Computer-Aided Logic Dasisn) koja sadrži etenderdne programa za funkcionalno, logičko pa čak i vrameneko modeliranje čvorova i elemenate na evim nivolma hijerarhije projektovanje Korišćenjem SCALDe vreme projektovenje Merka I je sa 100 godine-ljudi smenjeno na 2 godina-čoveka

Nije sve u elgoritmu

Aplikativni softver donoel eve afikasnije elgoritma i tako zahteva sve menji i manji broj korake da bi se došlo do željanog razultete Kade kupujate superkompjutar pazite de ime vame potraban eplikativni softver ili baram da vi znete keko ge možeta generisati ne herdvaru i elstamskom softveru koji kupujete

Koliki je udeo elgontma i eoftvera u brzini e koliki herdvere? Odgovor ne ovo pitenja je možde, gomalo iznanedujući e moža se dobiti gledanjem tabala 1 u kojoj ja dat razvoj brzine elgoritma za rešavanje trodimenzioneine eliptičke percijelne jednečine (to je tipičan zadetek ze jeden suparkompjuter) i letorijski razvoj i brzine nejbržih kompjutera u prošlosti Uporedite i denašnje računere se nakedešnjim superkompjuterime Biće vem jasno zašto je Nejl Linkoln iz CDC-e definisao suparkompjuter keo eletem koji je za samo jednu generaciju ispred računerskih potraba neuke i tehnika formulom

SUPERKOMPJUTER + KOLOSALNA IZRAČUNAVANJA + ENORMNA OČEKIVANJA + FANTASTIČAN RIZIK

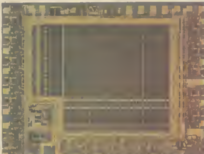
I zašto naslovi njegovih členeka glase Nije baš zabavno napraviti superkompjuter keo što je jednostavno projektovati ili Razvoj superkompjutera — zedovoljstvo i bol"

Sam algoritam nije dovoljan za brzu realizaciju na računaru ono što je brzo na jednom računeru, čak i ne vrlo sličnom može biti sporo Upečatljiv primar navodi Piter Benion program koji se izvršaveo brzinom od 2,4 megaflopa ne CDC 7600 Imao je, direktno pranet brzinu od 2 1 megaflopa ne Sajberu 205, što je preuređivanjem programe, povećano ne 110 megaflopova! (Šajber 205 ima 9 puta veću brzinu od CDC 7600 i proistekao je iz njage preko STARe 100 i Sajbera 203)

Oni koji su bili

Šte će nam biti dostupno u 1986 godini? Pogledejmo tabalu 2 Ze ljubitelje etarine dejemo podetke o suparkompjuterime iz 1983 pa ček i iz 1982 godine u tebeli 3

U tebeli 2 ete sigurno uočili goveću količinu znekove pitanje jer sam šturih



Sumrak silicijuma Nanoram na bazi galijum-arsenida kapaciteta 1 kilobit (256×4 bajle) ima vrame pristupa od svega 1 nanosekunde

Klejovskih (čitaj Krajovskih) izjave malo šta se pouzdenije zne o ovim kompjuterime, sem njihove hardverske osnove i činjenice da su sigurno proizašli kao i uvek do sade, iz prethodnih modele Pogledejmo zato, istoriju iz koje se ekstrapolecijom može zaključiti o onome što ćemo videti u ovoj godini

Sejber 205 je počeo de se realizuje početkom 1978 godine e usavršavenje u odnosu ne ranije modela se sastojelo u dodevanju bloka ze skalernu obređu podeteke i VLSI memorije koja ima korekciju avantuelnih jediničnih grešake i detekciju dvostrukih grešeke pri pranosu u ili iz nje Skalerni procesor ima 5 funkcionelnih blokova sa pokretnim zarezom, upravljački i esocijetivni blok Vaktorski procesor ime dve eritmetičke bloka i blok za komunikeciju sa memorijom U ekalarni procesor podeci mogu de ulaze evekih 20 nanosekundi, e u vektorski evekih 40 Sejbar ime 12 kanela za rad sa periferijeme i mogućnost izdavanje i primenje podeteke brzinom od 50 megabite u sekundi Prva mašine je postavljene i puštene u rad novembra 1980 godina CDC FORTRAN ne Sajberu ima posebne potprograme za pozivenje vektorskih operacije Cene je do 7,9 do 16 5 milione dolera.

Cray-1 ime mekeimelnu brzinu od 160 Mflopova, vrame sata od 12 5 nenosekundi i košta oko 4 5 milione dolera Proistekeo je iz CDC6600/7600 mašine koje je Kraj projektovao dok je u Minsapolisu radio za CDC Cray X-MP je poboljšan teko da je dostigao performanse iz tebele i u stveri koristi dve Cray-1 mašine U prodaji je i verzije X-MP sa četiri Cray-1 računera Cene od 5 do 14 milione dolera

Hitechi je leneirao S610/20 koji koristi prošireni M260H (računer kompetebilen sa IBM-om) i radi sa jednom do pet vektorskih jedinica O kvelitetime Fujitsu-ovog VP-200 dovoljno ja reči da ja u evim testovima amaričkih vrhunekih računekih institucija pokazao bolja performanse od Cray X-PM-e

Kraj se odvojio od CDC (Control Data Corp) e Amdal (Amdahl) od IBM Kod Amdala se mogu kupiti dva vektorska procesora modeli 1400 i 500 koji imaju brzinu

Tabela 1

algoritmi	*godina*	*broj operacija*	*broj operacija za n=1000*
Gausova eliminacija	*1945*	*n**7*	*10**21*
SOR iteracija (suboptimalna)	*1954*	*8n**5*	*8*10**15*
SOR iteracija (optimalna)	*1960*	*(8n**4) log(n)*	*8*10**13*
ciklična redukcija	*1970*	*(8n**3) log(n)*	*8*10**10*
multigrid metode	*1978*	*60n**3*	*6*10**10*

kompjuter	*godina*	*megaflopova (prosečno)*
Manchester Mark 1	*1947*	*0 0002*
IBM 701	*1954*	*0 003*
IBM Stretch	*1960*	*0 3*
CDC 6600	*1964*	*2*
CDC 7600	*1969*	*5*
Cray-1	*1976*	*50*
cOpCray-2	*1985*	*125*

od 1140 odnosno 133 megaflopova i memorije je od 256 odnosno 128 megabajte Dok je model 500 moguće nabaviti po ceni od 4 9 do 9 7 miliona dolara za model 1400 treba sačekati sredinu godina spramiti između 12 8 i 21 8 miliona dolara i naručiti ga na adresu Amdahl Corp P O Box 470 Sunnyvale CA 94086 USA O svim uslovima kupovine, ako vam se žuri možete se raspitati na telefon (408) 746-6000

Superbrzi logični niz Sprežna kola na bazi ECL tehnologije unose kašnjenje od samo 150 pikosekundi

i oni koji dolaze

Možda najlepši i najinteresantniji projekti u vezi sa kompjuterima visokih performansi vode se na američkim univerzitetima (na Kalifornijskom tehnološkom institutu Cosmic Cube Data Flow na MIT-u Pyramid na Ilionis univerzitetu u Urbani-Sempanj WISPAC na Viskonsin univerzitetu RISC na Stenfordu i Barkliju na Perdjuu kompjuter za obradu slika i mnogi drugi) U Evropi je posebno uspeo projekat Data Flow kompjutera na Menčesterskom univerzitetu u Engleskoj

Pored američkih i japanskih superkompjutera na tržištu će se uskoro pojaviti i rezultati britanskog Alvey projekta Esprit projekta Evropske komisije za komunikacije i nemačkog nacionalnog projekta U Kini je montiran Cray-1 a realizovan je i kompjuter sa preko 30 megaflopova

U okviru evropskog projekta Eureka francuska Matra i norveške mini kompanije Norsk data dobile su ugovor sa francuskom vladom za izgradnju superkompjutera brzine 100 Mflopova koji treba da bude završen do 1988 godine Francuski Bull i nemački Simens treba do 1990 godine da završe znatno moćniji računar

Ako posle svega ovoga niste mogli da se odlučite šta da kupite a treba vam stvarno stručan savet preporučujemo vam dva vrhunska stručnjaka za superkompjutere Miloša Ercegovca i Danijela Gajskog Ercegovac je šef Computer Science Departmenta na UCLA univerzitetu u Los Anđelesu a završio je elektrotehniku u Beogradu Gajski je profesor na Ilionis univerzitetu u Urbani Sempenj a elektrotehniku je završio u Zagrebu Obojica su doktorirala u SAD-u Ercegovac je primera radi predavač na predavanjima o superkompjuterima koja IEEE organizuje širom Amerike

Najbolji mikro-arhitekta sveta Tvorac „kreja-1" i „kreja-2" Sejmur Krej

Tabela 2

	CRAY 2	CRAY 3	ETA GF10	DENELCOR HEP-2	NEC SX 2
PERFORMANSE					
MAKSIMALNI BROJ MFLOPOVA	1000	?	10000	4000	1300
vreme sata (ns)	4	1	?	12.5	6
MAKSIMALNA MEMORIJA (miliona bajta)	?	?	256	4000	256
TEHNOLOGIJA					
logička kola	ECL	ECL(GaAs)	CMOS	ECL	?
glavna memorija	MOS	MOS	MOS	MOS	64 K MOS
HLAĐENJE	tečna	?	***	vazduh	statička voda

Tabela 3

	FUJITSU VP 200	HITACHI S-810/20	CRAY 1M	CRAY X-MP	CDC 205
PRVA INSTALACIJA	nov 1983	nov 1983	sep 1983	jun 1983	feb. 1982
PERFORMANSE					
maksimalni broj mflopova	500	630	250	630	400
vreme sata (ns)	15/7,5	15	12,5	9 5	20
VELIČINA GLAVNE MEMORIJE (milioni bajta)	256	256	32	32	64
TEHNOLOGIJA					
logičkih kola	ECL	ECL	ECL	ECL	ECL
glavna memorija	MOS	MOS	MOS	4K ECL	MOS
HLAĐENJE	vazduh	voda	freon	freon	freon

Miodrag Potkonjak

Kako postati milioner (2)

hakerski marketing

Šta ne programirati

Da vidimo pre svega šta nije lakše uraditi pomoću programa za računar.

(1) Ne treba pisati program koji će se izvršavati jedanput ili dvaput godišnje. Za pisanje programa potrebno je ručno napraviti jedan ili više test-primera i napisati program (tj. u opštim terminima izraziti problem i njegovo rešenje), a to znači da ćemo tokom stvaranja programa problem ručno rešiti najmanje dvaput. Kada se tome dodaju neizbežne greške u toku procesa planiranja, kodiranja i testiranja programa — onda postaje jasno da nema smisla pisati program koji se neće izvršavati bar tri-četiri puta godišnje.

Sledeće pravilo je, međutim, važan izuzetak.

(2) Treba napisati program koji omogućava da se uradi neki težak „pešački" posao pa makar i jednom godišnje. Ovo je jedan od najvažnijih razloga za široku prihvaćenost računara. Mnogi naši inženjeri kupili su računar da bi svoja standardna proračuna vršili lakše i brže, a studenti su prigrlili programabilne ručne kalkulatore kao izvanrednu pomoć na pismenim ispitima.

(3) Ne treba pisati program ako se dati posao može obaviti ručno za manje od pet minuta. Računari i njihovo programiranje mogu ponekad da iskomplikuju život. Nema baš mnogo smisla pisati program koji odgovara na pitanje „Kolika će mi biti plata ako dobijem povećanje 12.5%?", jer se to može izračunati napamet ili pomoću kalkulatora pa čak i pomoću računara u direktnom režimu rada. Sve nas to navodi na sledeće pravilo.

(4) Ne treba pisati program ako se cilj može dostići pomoću ručnog kalkulatora, kutije podsetničkih listića, ili pomoću običnog kalendara i olovke. Ovu vrstu „knjigovodstvenih" poslova kućni računari komplikuju učitavanjem programa pa morate da imate računar pri sebi itd. Pre nekoliko godina baš ovakve primene su navođene kao idealne za kućne računare ali to uglavnom nije tačno. Neki fizički medijumi za čuvanje podataka jednostavno nisu praktični i najbolji program za tekući račun ostaje nekorišćen na računaru koji ima samo kasetofon kao spoljnu memoriju.

(5) Ne treba pisati program ako tačnost nije bitna. Čemu podatak sa ekrana da je za tortu potrebno tačno 328 grama šećera ako se količina meri supenom kašikom a ne apotekarskom vagom?

(6) Ne treba ponovo izmišljati točak! Treba se raspitati da li se neki postojeći program može iskoristiti? Ako problem zahteva čist numerički proračun nad nizom matrica, onda je bolje nabaviti program za unakrsna izračunavanja (spreadsheet) nego mučiti se na bejziku. Neracionalno je praviti adresar kao zaseban program, spisak knjiga u kućnoj biblioteci kao zaseban program, sopstvenu bibliografiju kao zaseban program jer sve to treba da radi jedna dobra baza podataka.

Poznavanje bibliografije gotovih potprograma je od posebnog značaja. Na primer, za jezik C nude se posebne biblioteke za rad sa bazama podataka, za grafiku, za rad sa ekranskim prozorima, za konverziju bejzik programa u C-programe, i tako dalje. Na fortranu postoji ogromna biblioteka gotovih numeričkih potprograma, svaka knjiga iz paskala sadrži konane procedure i funkcije, a računarski časopisi takođe mogu biti značajna riznica tuđih iskustava.

Gornji kriterijumi pomažu da u želji za programiranjem ne zalutamo u pisanje trivijalnih programa. Razmotrimo sada na šta treba obratiti pažnju prilikom izbora teme za sledeći program.

Vera u srećnu zvezdu

Osnova za pisanje uspešnog programa je pravilo koje se često navodi u knjigama na temu „1001 način za brzo bogaćenje", a ono glasi: treba otkriti jednu ljudsku potrebu, a zatim joj udovoljiti. U skladu s tim treba prvo ispitati tržište, pa proceniti svoje mogućnosti.

Istraživanje tržišta je bitan deo priprema za pisanje novog programa. Na prvi pogled izgleda neobično, čak i nepotrebno ispitivati tržište mikro-računarskog softvera. Najveći broj programa pišu ljudi koji računare koriste za sopstvene potrebe, a ne kao medijum za razmenu i prodaju softvera. U programu za ličnu upotrebu kozmetika u vidu menija, HELP-naredbi, lepog štampanja i sličnog jednostavno nije ni potrebna. Ipak, program koji napišete za sebe u najvećem broju slučajeva ne biva uništen, naprotiv, često ostaje u svakodnevnoj upotrebi. Takav program primete drugi programeri i korisnici, zainteresuju se, zamole za kopiju i time sve počinje iznova: još jedan nepažljivo sročen program ulazi u svakodnevnu upotrebu. Srećom, postoje i pozitivni primeri, poput programa Sidekick napisanog za IBM PC i klonove. To je neka vrsta telefonske sekretarice za profesionalne programere: pamti sastanke, prekida rad računara u određeno vreme, čuva privatnu telefonska brojeve, radi i kao ručni kalkulator i slično. Program je nastao iz čisto privatnih potreba programera u Borlandovoj kompaniji (proizvođač Turbo Paskala), koje su bili često prekidani telefonima, sastancima i sličnim. Tek posle nekoliko meseci intenzivne upotrebe „unutar kuće", firma je primetila da u rukama ima pravo komercijalno blago.

Ako ste zakleti tvrdica i svoje programe nikom nikada nećete davati ni prodavati, čak i tada ima smisla praviti ljubazne programe. Možda ćete se datom programu vratiti kroz nekoliko godina, a tada sleduje neminovno čuđenje u stilu: „čekaj, čekaj, šta sad treba da radim, samo da se setim..." I u slučaju da ste vi svoje sopstveno „tržište", treba se potruditi oko početnog dizajna programa.

Računarska industrija još uvek boluje svoje dečije bolesti. Mnogi profesionalni programi započeli su svoj život bez skoro ikakvog istraživanja tržišta. Međutim, tržište softvera se izuzetno brzo menja, pa će uvek biti mesta za razvoj programa bez sigurne vizije o plasmanu na tržištu, samo sa verom u sopstvenu srećnu zvezdu.

Programiranje za budućnost

Osnovno pravilo istraživanja tržišta je do krajnosti jednostavno. Treba samo naći odgovor na pitanje „Šta ljudi žele?" Ovde razmatramo stvaranje uspešnih računarskih programa, pa to opšte pitanje pretvaramo u „Koje oblasti primene personalnih računara znam dovoljno dobro da mogu da ocenim šta je ljudima potrebno i šta žele da im se napravi?" U slučaju programa za slobodnu prodaju, pitanje se mora postaviti unekoliko drugačije: „Šta to ljudi hoće i mogu da plate?" Sva tri pitanja potenciraju da stvaralac programa pre pisanja programa mora da poznaje svoje buduće kupce. Akcenat je na reči „budući". Za stvaranje dobrog programa potrebno je nekoliko meseci ljudskog rada. Ako danas započnete svoje remek-delo, ono će biti spremno kroz godinu dana, dakle, morate oceniti šta će tada biti aktuelno na tržištu. Ne treba procenjivati sadašnje tržište — tu nemate šta da tražite, već tržište koje će tek postojati! Ovo je lakše reći nego učiniti, ali je baš to ključ uspeha.

Ima mnogo načina da saznate nešto više o svojim budućim kupcima. U slučaju da ste svoj sopstveni „mušterija" i da pravite program iz oblasti svoje specijalnosti, možda ste vi upravo tipičan predstavnik cele jedne kategorije stručnjaka? Najbolji specijalizovani programi iz npr. inženjerskih oblasti ili statistike stvoreni su upravo na taj način. Školski primer je nastanak programa Visicalc, odnosno programa za unakrsna izračunavanja tabela uopšte. Na tu ideju je došao jedan student ekonomije (!) da bi sebi olakšao predviđanje uspeha nove tržišne strategije kompanije Coca-Cola.

Život se komplikuje ako nameravate da pišete programe za komercijalnu upotrebu, odnosno prodaju. Možda za nameravani program već postoji tržište, a vaš program treba da bude u jednom ili više detalja bolji od postojećih? Tada je bitno da saznate koliko je sličnih programa već prodato, koje su prednosti na strani konkurencije

Posle izvesnog vremena provedenog uz računar, bejzik i ostali programski jezici postaju sredstvo za dostizanje nekog cilja. U prvo vreme uspeh je napisati program koji radi. Kasnije, kodiranje, tj. ispisivanje naredbi na programskom jeziku, postaje stvar rutine, a u prvi plan izbija svrha programa. Postoje dva osnovna pravca produktivnog iskorišćenja te razuđene intelektualne snage: ne treba pisati trivijalne programe, odnosno treba dobro proceniti kakve su potrebe za novim programom kojeg nameravate da napišete. Bogatstvo se teško može steći pisanjem programa kakve biste sami voleli da imate, već samo pisanjem programa koji su potrebni velikoj armiji korisnika.



Ilustracija: Miša Marković

kakav je trend u razvoju softvera, kakve su bile kritike po časopisima itd. Osim primarnog, da li postoje i sekundarna tržišta, a zatim da li se tržište u Evropi razlikuje od tržišta u SAD, i slično; koliki obim tržišta nameravate da obuhvatite da biste bili zadovoljni, a recimo koji deo tržišta pokriva vaše troškove, i tako dalje.

Mogućnosti procene su znatno sužene ako vaš program predstavlja stvarnu novost na tržištu. Bez pravih informacija, možete podeliti potencijalne mušterije u dve klase. U prvoj su svi oni koji bi kupili kompletan računarski sistem samo zbog vašeg programa ili zajedno sa njim. Najpoznatiji primer te vrste je Visicalc, za koji kažu da je prodao nekoliko stotina hiljada računara Apple II. Danas je slična situacija sa programom Lotus 1-2-3 i računarima tipa IBM PC, a na kućnim računarima postoji primer avanture Elite zbog koje je prodat izvestan broj računara BBC model B, dok bi se slično moglo reći za Hu-BASIC u odnosu na računar Sharp MZ-700/800.

U drugoj klasi su svi kupci koji već imaju neki računar, pa bi za njega hteli da kupe vaš program. Zatim možete da uporedite cenu vašeg proizvoda sa prihodima potencijalnih kupaca i da procenite koliko vaš produkt može da im uštedi ili donese para (direktno ili indirektno). Na osnovu svega toga možete da utvrdite kojoj grupi želite da ponudite vaš program i po kojoj ceni.

Kako prodati program

Nije dovoljno imati kvalitetan produkt — treba ga na pravi način i reklamirati. Marketing je tradicionalno slaba tačka računarske industrije, mada je otprve jasno da niko ne može da kupi vaš program ako i ne zna za njegovo postojanje. Osnovni princip marketinga je zavodljivo jednostavan: treba privući pažnju na svoj proizvod, a zatim je tu i održati. Dva su standardna načina za to: reklama preko novina i specijalizovanih revija, odnosno reklama kroz prodavnice koje će vaši potencijalni kupci posetiti. Trenutno u Jugoslaviji ne postoji široka mreža prodavnica koje bi bile isključivo posvećene računarima. Ako se u prodavnici muzikalija prodaju i računarske kasete, onda bi bilo dobro oformiti specijalne štandove tako da računarske kasete budu odvojene od muzičkih, napraviti prospekte sa kratkim opisom programa koje bi radoznali kupac mogao da ponese kući i da se na miru odluči, zatim atraktivno pakovanje sa npr. pravilima igre ili osnovama dokumentacije za program, a i da ne spominjemo demonstracije programa na računaru uživo. Uzgred, među velikim softverskim proizvođačima sve prisutniji je trend prodaje demo-verzije programa, tako da kupac za

znatno manje para može da se iz prve ruke uveri šta program radi

Uspeh programa bitno zavisi od kvaliteta prateće dokumentacije. Rad programa se obično prikazuje kroz slike ekrana. Nije lako napisati dobar priručnik. Problem je u tome što je krajnjem korisniku svejedno kako program radi „iznutra" — ako dobro radi. Dokumentaciju prečesto pišu sami pisci programa, a oni na svoje čedo gledaju pristrasno i kroz priručnik demonstriraju briljantno poznavanje računara, mašinskog jezika i slično. Moglo bi se pomisliti da je odsustvo bilo kakve dokumentacije najgora vrsta dokumentacije, o čemu bi kupci piratskih programa sigurno imali šta da kažu. Međutim, od nepostojeće dokumentacije još je gora netačna dokumentacija koja se često sreće kod novih mašina na tržištu.

Dokumentacija treba da se sastoji iz najmanje dva vrsta priručnika. Vašem kupcu je prvo potrebna početnica za korišćenje programa, dakle uputstvo kako da odmah uz pomoć programa počne da radi nešto konkretno. Korisniku u toj fazi nisu potrebne dugačka lista naredbi niti šta sve program može, već jednostavno upoznavanje sa mogućnostima programa. Druga vrsta priručnika je podsetnik, u kojem su sve moguće opcije logično povezane. Detaljan sadržaj i indeks su od velike važnosti za ovu vrstu priručnika.

Čarobni trougao

Računare često koristimo kao zamenu za nedostupnog stručnjaka. Program za tekući račun bi trebalo da radi poput knjigovođe — vrhunskog poznavaoca bankarskog poslovanja. Svaki korisnički program u sebi mora da sadrži neko znanje koje se ispoljava u vidu dijaloga programa sa krajnjim korisnikom. Još uvek ne postoje programi koji pišu druge programe bez pomoći čoveka, a krajnji korisnici računarskih programa i dalje su ljudi. Zato stvaranje dobrog programa u sebi mora objediniti znanje i veštine triju profila: eksperta, programera i korisnika. Odnos ovih saučesnika u pisanju programa je šematski prikazan na slici 1. Brojevi od 1 do 4 označavaju različite interakcije između ova tri osnovna tipa.

Ekspert, na primer lekar ili advokat, poseduje duboko poznavanje principa svoje struke, što je većim delom irelevantno za strukturu programa. Programer ne mora da poznaje te principe, ali je u stanju da kodira veštinu eksperta na način dostupan računaru, a krajnjem korisniku potreban je samo deo te veštine — upravo ono što program i treba da radi. Slično, programer poseduje opšta znanja iz softverskog inženjeringa (tj. zna da napravi program na bilo koju zadatu temu), a korisnik zna specifične uslove u kojima će se program izvršavati, ali su te „konkretne situacije" po pravilu nepoznata i programeru i ekspertu.

Interakcija između pojedinih partnera je takođe trojaka. Ekspert mora programeru da prenese svoje znanje i to se obično naziva sistemskom analizom (na slici 1 označeno brojem 2). Taj korak uvek prethodi samom progamiranju i prilika je da se demonstrira komunikacioni jaz između modernih profesija. Cela kibernetika kao nauka je i nastala iz želje Norberta Vinera da se uspostavi efikasna razmena informacija između stručnjaka različitih profesija. Problemi komunikacija se obično razrešavaju tako što programer nauči delić neke druge nauke, a ekspert sazna ponešto o programiranju, što sve čini programerski posao tako interesantnim. Moderniji način da se preskoči ovaj jaz je korišćenje tzv. ekspertnih sistema, no o njima nekom drugom prilikom.

Preklapanje između korisnika i programera spada u probleme „računarske konverzacije" (označena brojem 4). To je zapravo način na koji će se budući program obraćati korisniku, a već sada je jasno da se upravo na ovom polju bije odsudna bitka za definitivno prihvatanje računara u svakodnevnom životu. Ne moramo da znamo kako radi gramofon niti kako je snimljena neka ploča. I nadajmo se da jednog dana nećemo morati da znamo ništa o principima rada operativnih sistema, a da ipak budemo u stanju da koristimo računar poput gramofona, telefona, televizora i svih ostalih čuda popularne elektronike.

EKSPERT
znanje
veština

KORISNIK
konkretne
situacije

3 1 2 4

PROGRAMER
softversko
inženjering

LEGENDA

Preklapanje između eksperta i korisnika (broj 3 na slici 1), obično se previđa u knjigama o računarskom programiranju jer se one uvek koncentrišu na radnje bliske računarima. Daleki, nedostižni cilj računarske industrije je da sva znanja postanu dostupna kroz konverzaciju sa računarom.

Konačno, zajednički deo za sva tri profila je računarski program (broj 1 na slici 1). Uloga programera bi trebalo da bude pasivna: on mora da nauči kako bi želeli da sarađuju ekspert i korisnik, i nije njegov zadatak da nešto menja u interakciji između njih. U pisanju korisničkih programa programer je najslabija karika u lancu!

Sve u svemu, poznavanje programiranja samo za sebe nije dovoljno da bi se došlo do dobrog korisničkog programa. U nedostatku ekspertnih znanja, hakeru preostaje da piše igre za uništavanje džojstika ili avanture — ili da se bavi programiranjem radi sopstvenog intelektualnog napretka. Osećaj neusmerenosti koji se javlja kod uznapredovanih početnika treba prevazići ličnim uzdizanjem i produbljavanjem znanja u izabranoj oblasti. Time dolazimo na već spomenut, ali na sreću relativno čest i plodotvoran slučaj, kada su sva tri profila — ekspert, programer i korisnik — spojeni u jednu ličnost.

Duško Savić

Šta ima novo

Bibliografija računarskih knjiga

Kupujete sve knjige iz računarstva, zar ne? Da biste bili sigurni da vam neka bar od američkih izdavača nije promakla, nabavite Annual Bibliography of Computer Oriented Books koji izdaje Kolorado univerzitet (to je onaj što se tu i tamo, vidi u Dinastiji, a vi verovatno niste baš naj-sigurniji koja je to zgrada). Knjige su podeljene u 86 kategorija prema tipu (priručnik, udžbenik, monografije) i stilu izlaganja (programirani kurs, naracija, esej), a najviše ih je iz oblasti personalnih računara (kao, uostalom, i kod nas). Cena kataloga je 4 dolara a adresa na koju možete da uplatite ovu svotu je Computing Newsletter Box 7345, Colorado Springs CO 80933 USA. Požurite, jer katalog se stalno osavremenjuje, tako da se knjige starije od 3 godine brišu sa spiska.

N. Aleksić

Za male Francuze

Francuska vlada je razdelila oko 120 000 računara osnovnim školama u Francuskoj. Paralelno sa tom velikom deobom 110 000 učitelja je pohađalo blic kurseve obučavanja za rad na računaru. Mi ćemo još pisati o ovom zanimljivom (to je najmanje što se može reći) pokušaju, a vi razmislite koliko sve ovo liči na situaciju kod nas?

Vremena su konfuzna

Sud je zabranio da se Somašild (Somashield), specijalne folija koja se lepi preko ekrana monitora, reklamira kao sredstvo koje smanjuje štetno dejstvo zračenja ekrana zato što ne postoji pouzdani dokazi da su ekrani zaista štetni. Zaista?

Enciklopedija na disku

Američka firma Grolijer (Grolier) je stavila svih devet miliona reči Američke akademske enciklopedije na jedan jedini kompakt-disk prečnika pet palaca. Disk se može koristiti kao ROM, ali samo za IBM PC i „atari 520 ST". Na disku se, pored enciklopedije sa 30 000 tema (koja, inače, staje u dvadeset jednu knjigu) nalazi i potpuni indeks reči i softver koji omogućuje baratanje. Sasvim je za verovati da bi jedan ovakav disk bio dovoljan i za osam naših enciklopedija. Ko zna, možda nam je stvarno toliko i potrebno.

Računari iz mog ugla

RAČUNARI KAO POLITIČKO PITANJE

Ilustracija Miša Marković

Ako imate bar malo igračkog staža u arkadnim igrama mogli ste da primetite da često viši nivoi nisu teži od početnih. Jedino što treba nekako dospeti do tih nivoa.

Na prvi pogled izgleda da ni bavljenje računarskom politikom na visokom nivou nije osobito teško. Osim što nema pravo svako da nastupi na tom nivou.

Kompjutarski strateg pojavljuje se na TV u odelu tamnijih boja. Važno je da odabere dostojanstvenu pozu da mu glas bude siguran (po mogućstvu dubok) i uopšte da ostavlja autoritativni utisak. I naravno važno je da pravilno odabere đezen na svojoj kravati koji će dovoljno dobro održavati njegovu ličnost. Ako se upravo vratio iz Amerike može otkopčati dugmad na kariranom sakou koji je druge boje od besprekorno opeglanih pantalona. Povremeno će ležerno staviti ruke u džepove.

Dok to gledate vi lečite svoj kompjutarski kompleks loveći autoritete u greškama. Kad ustanovite kako oni bulazne vama će svakako biti mnogo lakše. Nemojte davati paušalne ocene da je jednostavno biti računarski političar i da je dovoljno biti ozbiljan klovn. Nije lako biti u nekom inicijativnom odboru za standardizaciju računara ako prema malim računarima osećate averziju do te mere da ne možete ni da ih pogledate. Među njima ima vrlo dragocenih ljudi koji nisu čuli da kod nas postoje računarski časopisi. Samo oni mogu bez predrasuda i predubeđenja objektivno da kreiraju računarsku politiku.

O računarskoj strategiji ne treba suditi na osnovu površnog posmatranja računarskih stratega. To je odgovorna, složena i kompleksna problematika koja se ne može svesti na prvu aproksimaciju. Nikako nije cilj da računari budu što praktičniji sa izobiljem softvera, i što jeftiniji. Obični ljudi svojom prostonarodnom logikom nisu u stanju da dokuče svu suptilnost računarske politike.

To što su kod nas školske vlasti jedne republike dale preporuku da tasteri RETURN i RESET treba da budu jedan ispod drugog a oni za pomeranje kurzora sa leve i desne strane tastature nije glupost. Sve to ima dublji didaktički smisao. Recimo ako ste gađali RETURN pa malo promašili i resetovali računar to će vas naučiti da budete pažljivi i da olako ne pritiskate dugmiće.

Pomeranje kurzora čak prevazilazi didaktičke okvire i ima širi društveni značaj. Ako su tasteri raspoređeni prema preporuci nećete moći da pomerate kurzor samo jednom rukom. Ovakvim uravnoteženim simetrično-skladnim rasporedom primoravate učenike da obe ruke drže na računaru a ne da tamo nešto mute ispod klupe. Trebalo bi tu koncepciju razviti i dalje, da za korišćenje računara bude potrebno tri četiri pa i više ruku. Tako će više učenika da radi na istoj mašini što će razvijati njihov kolektivni duh a onemogućavati uvek opasne individualiste koji bi nasamo mogli da zloupotrebe računar. U vezi s tim, predlažem tastaturu se osam tastera od kojih bi jedan bio univerzalan (za sve znakove) i sedam (7) raznih SHIFT-ova. Trebalo bi razmotriti i predloge naših čitalaca ukoliko ih ima.

Obilje jeftinog softvera nije prednost nego velika opasnost. Ne samo da time prepuštamo omladinu pogubnim uticajima nego se i moralno izopačuju preuzimavajući programe. Pod hitno bi morala biti uvedena komisija koja bi pregledala i cenzurisala softver i institucija koja bi potpuno preuzela na sebe obaveze i odgovornosti distribucije. Cene nikako ne smeju da budu niske. Kome treba taj će i da plati.

Nije cilj da računari budu što jeftiniji. Učinjena je greška i dozvoljeno je da narod stihijski nabavlja računare i da za to troši dragocene devize koje bi lako dobrodošle domaćim proizvođačima. Ljubitelji inostrane robe morali bi imati na umu englesku poslovicu: „Nisam dovoljno bogat da bih kupovao jeftine stvari". Mnogo je važnije da se naša opštinska ekonomija celovito i kompletno razvijaju.

U osnovi zdravu ideju „u svakoj opštini omladinska fabrika" koja bi proizvodila računare potpuno nekompatibilne sa drugim, trebalo bi proširiti i po mesnim zajednicama otvarati penzionerske fabrike. Tako bi se i penzioneri držali na oku jer su nešto počeli u poslednje vreme mnogo da gundaju. Ujedno bi se zadovoljio princip da neko ne treba da radi ono što mu se ne sviđa.

Ovako krupne strateške greške ne bi bile počinjene da smo na vreme mislili i školovali odgovarajući kadar. U škole svakako treba uvesti usmerenje kompjutar-političar. Budući kreatori naše svetle računarske budućnosti treba da kanališu računarstvo da posluži očuvanju stanja kakvo jeste. To će biti vrlo teško, s obzirom na nizak nivo društvene svesti privatnih lica koja odbijaju da kupe bar neki od domaćih računara. Moguće je da ubeđivanja neće uroditi plodom jer postoji veliki broj zlonamernih, potpuno nekompletnih pojedinaca (ne razumeju se ni u računare ni u strategiju) koji vrše poređenja domaćih i stranih računara kao da se to može porediti.

Ako drugačije ne ide trebalo bi iz osnova promeniti globalnu strategiju. Trebalo bi se opredeliti za jedinstveni računar za celu zemlju. Zgodno bi bilo da se usvoji neki model koji niko više nigde ne proizvodi i koji može biti kompatibilan samo sa običnom peglom. Kada bi to bilo usvojeno kao standard naša bi industrija bila zaštićena od nelojalne konkurencije. Naravno da bi tada i računarske novine morale da prestanu da pišu o stranim računarima da se javnost ne bi bez razloga uznemiravala i da bismo bezbrižno zakoračili u 21. vek.

Jelena Rupnik

Kako to rade drugi

piratska je tuga pregolema

Pirati su postali omiljena tema u svim računarski orijentisanim glasilima. Malo ko je propustio šansu da piše o toj temi u kojoj se ukrštaju interesi svih učesnika računarskog tržišta. Kako vreme prolazi, rasprava o piratima se polako zaoštravaju, a velike firme prelaze sa reči na dela. Odnosi su sad toliko konfrontirani da je direktan sukob neminovan, a kako taj sukob može da otpočne pokazuje upravo ovaj tekst. U našoj zemlji, praktično, i ne postoji tržište softvera, pa domaći pirati za razliku od američkih, još uvek mirno dišu. Filmski distributeri su uspeli da izađu veoma brzo na kraj sa piratima video kaseta. Da li to znači da se i piratima kompjuterskih programa sprema slična sudbina?

Pirati su imeli sreće dok su operisali u „dobrim, starim vremenima" kada su ih jedni tapšali a drugi im pretili prstom a oni samo uzimali — lovu. Sad je sve sasvim drugačije. Velike kompanije su shvatile neke stvari i naravno „poduzele neke korake". Koje?

U najvećim računarskim časopisima širom SAD objavljen je oglas koji izgleda ovako. Na slici je mlad, lepo obučen čovek (odelo i kravata). On je sasvim nalik uspešnom poslovnom čoveku — u ruci mu je telefon, a lice mu je nasmejeno. Pored njega piše:

„Običan kriminalac

On ne izgleda kao običan kriminalac ali on je upravo protivzakonito prekopirao jedan program. To nije običan zločin. To je zločin saveznog nivoa.

Kopiranje softvera bez dozvole je kršenje zakona o autorskim pravima i kažnjava se po zakonu, znači lišavanjem slobode i novčano.

Ovom kriminalcu ni slučajno ne bi palo na pamet da ukrade novac iz nečijeg novčanika. On je čovek od principa. No, on ne shvata da je pravljenje protivzakonite kopije programa identično sa džeparenjem autora programa. On takođe ne shvata da svi plaćaju zbog njegovog čina, jer će oni koji program kupuju platiti višu cenu.

Ako poznajete osobe koje protivzakonito kopiraju programe recite im da krše zakon.

Pomozite u zaustavljanju ovog kriminala pre nego što bude kasno.

NEOVLAŠĆENO KOPIRANJE SOFTVERA JE KRIMINAL!"

Dileme nema: proizvođači softvera preko svog udruženja objavljuju krstaški pohod na pirate.

Izgleda da je udela u ovom potezu imala i velika anketa na temu piratstva. Anketa je sprovedena na uzorku od 45000 ljudi i pokazala je da se na svaki legalno prodati program proda i jedan piratski. Suočeni sa takvim podatkom, proizvođači su morali da „uzvrate udarac". Čak ni sa pretpostavkom da svi oni koji su program kupili od pirata ne bi kupili taj isti program legalno situacija ne izgleda mnogo bolje. Procene za 1986. godinu su da će na celokupnom tržištu softvera 25 do 30% prometa ostvariti pirati. Gubitak u zaradi za softverske firme je reda veličine 800 miliona dolara.

Uvežbavanje otpora

Softverski lobi u Kongresu SAD je boksovao zakone koji bi trebalo da štite proizvođače softvera, a prva primena federalnog zakona upravo se isprobava na tajvanskom biznismenu Teh Yi Huangu koji se bavio piratstvom velikog obima, a uhvaćen je na delu. Njega sad očekuje kazna zatvora do 14 godina trajanja (!!!) plus troškovi suđenja koji idu i do 75000 dolara.

Osim iznuđivanja ovakvih zakonskih mera, udruženja organizacija koja se bave obradom podataka postavila su i određena pravila koja treba poštovati pri poslovanju:

— Svaki zaposleni mora da potpiše izjavu da će čuvati softver od pirata kao i da je upoznat sa posledicama učestvovanja u aktu piratstva (posledice se kreću od otkaza do krivičnog gonjenja).

— Potrebno je razviti metode redovne kontrole cirkulacije, distribucije i kopiranja softvera kao i sastavljanje spiska zaposlenih koji imaju pristup softveru.

— Treba formulisati pravne metode koje bi kompanije primenjivale da zaštite sebe u slučaju piratstva. One treba da su tako formulisane da krivica i sve posledice padnu na onoga ko je i kopirao programe.

— Neophodno je postaviti kontrolne detektore na izlazu iz firmi — sličnih onima na aerodromima — da bi se smanjila mogućnost iznošenja raznih medijuma sa podacima.

— Pored ekipa za fizičko obezbeđenje zgrade od provala, požara i sličnog poželjno je formirati posebne timove čiji je jedini zadatak da štite softver kako od pirata tako od oštećenja, zlonamernog menjanja i sličnog.

— Predviđeno je osnivanje posebnih fondova koji treba da pokrivaju štetu koju naprave pirati (pre svega za nabavku novih programa umesto ukradenih). Pošto se iz razloga bezbednosti ne prave kopije, jedan isti program se kupuje od originalnog proizvođača softvera više puta.

— Personal treba obučiti da se štiti od pirata (putem kurseva, seminara, plakata i brošura, pri čemu je glavni slogan „Ne budite naivni!").

— Treba organizovano prići propagandi i izazvati diferencijaciju među piratima. Treba ih izdeliti na one koji to rade za novac, one koji rade iz radoznalosti, one koji to

rade sebe radi i tako dalje sa posebnim akcentom na kriminalnom aktu koji podleže zakonskim sankcijama

Nije ostalo sve samo na zakonskom gonjenju i ovim preporukama. Neke firme su probale i sa zaista originalnim rešenjima. Kompanija Mikropro internašenel (Micropro International) je pokušala da daje besplatno originalne programe za programe koji su kupljeni od pirata. Zauzvrat je tražena informacija o tome kod koga je program kupljen. Akcija je trajala šest meseci, na njoj su radila četiri službenika sa punim radnim vremenom. „Ali veliki bosovi i oni koji su krali programe na izvoru — nisu uhvaćeni." Na kraju je Mikropro internešenel digao ruke od jalovog posla i prestao sa akcijom. „Osećao sam se kao direktor Dženeral Motorsa (General Motors) koji daje nove automobile za ukradene", rezignirano je izjavio posle akcije direktor Mikroproa, „ipak bio je to dobar pokušaj."

Crvi u jabuci

U toku ogorčene borbe na videlo su isplivale i neke činjenice koje sugerišu da inicijalni krivci sede možda baš unutar softverskih i hardverskih firmi. Predstavljajući se kao spoljni saradnici firme Epl (Apple) dva privatna detektiva koje je unajmio jedan računarski časopis (na nivou Računari) su od jednog službenika firme za samo 1100 dolara kupili desetak vrlo skupih programa za računar „makintoš". (Među programima je bio i čuveni „Jazz".) Jedan od kupljenih programa je bio i program koji je u Eplu bio samo na doradi i proveri, tako da još nije bio pušten u prodaju (redovna je praksa da softverske firme šalju proizvođačima hardvera programe na probu pre izbacivanja na tržište).

Pritešnjen dokazima, službenik koji je programe prodao je izjavio da u Eplu vlada hakerska atmosfera još od osnivanja firme, da je kopiranje programa redovna praksa, da to „rade svi" i da je on do programa došao jednostavno tako što su mu drugi spustili kopije na sto. U firmi je već jednom bio uhvaćen i upozoren, pa je zato preneo kopije programa kući i nastavio celu rabotu tamo, ali samo za prijatelje Epla. Naravno, uz skromnu nadoknadu.

Nezvanično, jedan od direktora proizvodnje u Eplu kaže da je „Epl kao prodavnica bombona. Teško je odoleti iskušenju" te da se dobar program prenosi kroz firmu „kao požar" i „dok se okreneš svi ga imaju". Po njemu ipak treba praviti razliku između onih koji to rade da bi ostvarili nezakonitu dobit i onih koji to rade radi analize programa ili stručnog usavršavanja.

Zvanično, u Eplu su „iznenađeni i uvređeni". Tvrde da će se primeniti drakonske mere nad onima koji budu uhvaćeni u piratstvu „ali i da se, to dešavalo i dešavaće se".

Izgleda da intenzivirana borba protiv pirata sa jedne strane i otkrivanje legala pirata u najvećim računarskim firmama tek uokviruju ceo piratski problem i da rešenje nije ni malo blizu. Ono što je sigurno je da protiv pirata započeta bespoštedna borba sa ciljem da se oni u potpunosti istrebe.

Odgovor pirata još nije poznat. Da li će osnovati udruženja za zaštitu ugroženih pirata ili će jednostavno samo — povećati svoje cenovnike?

Ivan Nador

Biblioteka knjiga

Početnički za početnike

T. F. Fraj: Računari za početnike

Izdavač Nolit, tiraž 7 000 primeraka, cena 1 200 d

U Nolitovoj biblioteci „Zanimljiva nauka" pojavio se prvi naslov posvećen računarima — prevod knjige T. F. Fraja „Računari za početnike". U knjizi se kroz poglavlja Osnovni pojmovi o računarima, Brojčani sistemi i računar, Računarska logika, Računarski ulaz i izlaz, Centralni procesor, Uskladištavanje i iskladištavanje informacija, Kako funkcionišu računarski programi, Veliki i mali računarski sistemi i Računar i društvo, pokušava da približi širokom krugu čitalaca način funkcionisanja računarskih sistema.

Kao što iz sadržaja možete videti, obrađene su teme iz nastavnih programa svih naših srednjoškolskih usmerenja posvećenih računarstvu, kao i početnih kurseva programiranja na višim školama i fakultetima. Dakle, u ovoj knjizi govori se o stvarima o kojima već desetak godina postoji više knjiga na srpskohrvatskom jeziku naših autora. Stoga bi logično bilo očekivati da knjiga čija je svrha popularisanje nauke bude čitljivija, bolja i jeftinija od drugih koje tretiraju iste teme.

Mada autor ovog prikaza, koji već dosta godina predaje upravo te osnovne pojmove računarstva početnicima, nema puno suštinskih primedbi originalu, engleskoj verziji ove knjige, i mada smatra da ni jedna knjiga ne može biti štetna niti suvišna, ne može a da ne skrene pažnju potencijalnim čitaocima ove knjige na sledeće činjenice.

Po svemu sudeći, u Nolitu nisu smatrali da je potrebno da pri donošenju odluka koju knjigu treba izabrati za početnike u računarstvu konsultuju nekog ko se računarima i bavi. Ova greška ponovljena je i time što prevod izabrane knjige najverovatnije nije dat na stručnu recenziju. Kako inače objasniti da se u knjizi, koja bi trebalo da bude uzor kako uvoditi nove pojmove, demonstrira elementarno nepoznavanje stručne terminologije. Tako se umesto odomaćenog „spoljašnja memorija" koristi termin „skladište", prihvatna memorija (bafer) ovde je „tampon-spremište", a uvode se i „potporna skladišta". Oni koji su odlučili da svoje vidike prošire pomoću ove knjige mogu naučiti da se bajt sastoji najčešće od šest bitova (ne pominje se ni jedan drugi eksterni kod sem šestobitnog), kao i da „U decimalnom računu komplement jednog broja je onaj broj koji mu se mora dodati da bi se kao rezultat dobila nula". Nije razjašnjeno šta je potpuni a šta nepotpuni komplement, a koliko navedena definicija ima smisla, procenite sami.

Ovo je, na žalost, tek deo primedbi koje se mogu staviti na Nolitove „Računare za početnike". Istina, i u knjigama koje smo ranije prikazivali moglo se naći nedostataka, kao što i ova ima svojih dobrih strana. Pre svega, u njoj se može naći niz analogija sa načinima rešavanja problema iz svakodnevnog života koje na slikovit način opisuju kako računari funkcionišu. Uz mnoštvo drugih informacija dato je dosta primera primene računara. Sve u svemu, ova knjiga može biti korisna kao dopunska literatura kako onima koji tek upoznaju računare, tako i nastavnicima programiranja, ali ne bismo je mogli preporučiti kao jedini izvor saznanja o ovoj oblasti.

Razlog što smo sada znatno stroži je što se prevod ove knjige, koja metodološki nije bolja od odgovarajućih knjiga naših autora, našao u renomiranoj biblioteci renomiranog izdavača, dok su prethodne knjige bile prve koje na našem jeziku pokrivaju određenu oblast, prvenci svojih mladih autora, štampane u malim tiražima, te su im njihovi izdavači mogli i morali oprostiti za sitnije nedostatke. Ne možemo se oteti utisku da je zarada koja se može ostvariti od prodaje 7 000 primeraka knjige „Računari za početnike" za izdavača bila važniji motiv od popularizacije nauke. Žao nam je što se u izlozima naših knjižara uz ovo izdanje ne može videti i udžbenik sa sličnim, ali korektno obrađenim sadržajem — „Osnovi računarskih sistema" T. Katanića i M. Markovića koji je Zavod za obrazovanje administrativnih kadrova SR Srbije štampao u tiražu od svega 200 primeraka. Tada bismo, bez obzira na cenu, našim računarskim početnicima imali šta da preporučimo.

No nije to jedina knjiga koja vas na pristupačan način može uvesti u svet računara. Ako ste odlučili da saznate nešto o računarima i mogućnostima njihove primene, prelistajte pažljivo sve knjige koje se mogu naći u našim knjižarama tehničke literature, pa tek onda donesite odluku da li ćete kupiti baš onu koja je najnovija i koja se najviše reklamira.

N. Spalević

„amstradov" kernal

Operativni sistemi

Postoje tri različita vrsta programskih celina koje se mogu naći u „amstradovoj" memoriji. To su programi prednjeg plana (foreground), programi zadnjeg plana (background) i RSX (resident system extension).

Princip upotrebe bočnih ROM-ova kod „amstrada" gotovo je isti kao i kod BBC-jevih računara. Jedina razlika je u tome što „amstrad" pejdžuje ROM-ove preko adresnog prostora koji zauzima i RAM, i to od #C000 do #FFFF (tu se obično nalazi video memorija), dok je kod BBC-ja određeno 16K koji se koriste samo za ROM memorije. U praksi to izgleda ovako: jedan od ulazno-izlaznih portova procesora (#DFxx — bitan je samo bajt veće vrednosti) se koristi za određivanje da li će u poslednjoj četvrtini adresnog prostora biti uključen neki od ROM-ova ili RAM. Tako svaki od ROM-ova koji su priključeni ima svoju adresu od jednog bajta. Pri tome su četiri vrednosti rezervisane: 252 — uključi donji i gornji ROM, 253 — isključi donji i uključi gornji ROM, 254 — isključi gornji i uključi donji ROM i 255 — isključi donji i gornji ROM, i one se koriste za stanja ROM-ova, dok se ostale vrednosti koriste za biranje gornjih ROM-ova (tom prilikom stanja se ne menja). Donjih 16K ROM memorije je rezervisano samo za operativni sistem. Napomenimo da stanje ima uticaj jedino na čitanje sa određene memorijske lokacije koju zahvata preklapanje RAM-a i ROM-a, dok se upis uvek vrši u RAM. Ovo važi samo za procesor — video kontroler 6845 uvek ima pristup do RAM-a gde se smešta slika.

Na spoljni port opšte namene mogu se priključiti do 252 bočna ROM-a, od kojih je 7 sa programima zadnjeg plana, i to na adresama #01—#07. Posle inicijalizacije operativni sistem ispituje da li se na adresi =00 nalazi neki ROM i, ako se nalazi, kontrolu predaje prvom ulazu koji pronađe u tabeli naredbi ROM-a 0 (obično se i nalazi samo jedan ulaz), u suprotnom se proziva osnovni ROM (bejzik). Ostali programi prednjeg plana se mogu iz bejzika jednostavno pozvati jednom naredbom kao WORD (pokušajte jednom BASIC).

U prvoj liniji

Računar se nalazi pod kontrolom programa prednjeg plana, tj. njima je prepuštena komunikacija sa korisnicima i odluka o tome kakav odgovor dati na određeni zahtev. Takav je recimo „amsword", na kome sada nastaje ovaj tekst, ili tolikohvaljeni železničarski (Locomotive) bejzik koji je ugrađen u računar, kao i mnogobrojne igre. Ovi programi se mogu podeliti na dve podvrste: jedni su u ROM-u, a drugi u RAM-u. Iako sam operativni sistem među njima ne pravi razliku, sa stanovišta korisnik razlika je ogromna. Programe koji su predviđeni da rade iz RAM-a potrebno je stalno učitavati sa kasetofona ili diska, pored toga oni obično oduzimaju priličnu količinu slobodnog prostora. ROM forground program ili bočni (sindaways) ROM, kako se najčešće naziva, uvek je prisutan i zauzima mali deo RAM-a koji se koristi samo za promenljive i radni prostor. Karakteristično je da se u ROM upisuju programi koji se jako često koriste. To su uglavnom sistemski (programski jezici, asembler) i uslužni programi (tekst-procesori, programi za rad sa bazama podataka). Moraćemo da razočaramo mnogobrojne ljubitelje igara — svoju omiljenu igru neće moći da pronađu u ROM-u i jedino im preostaje da angažuju lokalnog hakera da je priredi u novom aranžmanu.

Programi prednjeg plana koji se nalaze u ROM-u mogu biti dugački do 64K. Pošto već naziramo pravu bujicu pitanja, hitamo da damo odgovor. Moguće je (softverski) povezati četiri ROM-a tako da u stvari čine jedan program. Prelazak iz jednog u drugi vrši se na vrlo jednostavan način. Jedan od restarta je žrtvovan da bi se omogućio poziv u drugi ROM sa samo tri bajta, dakle istom dužinom kao i normalna CALL instrukcija. To je RST #10, iza koga sledi 16-bitna reč, pri čemu gornja dva bita određuju u kom od ROM-ova se nalazi potrebna rutina, a donjih 14 bita određuje adresu relativno od #C000. Na taj način moguće je smestiti komplikovane i visoko profesionalne programe u ROM i opet imati na raspolaganju onih 40 i nešto kilobajta slobodnog prostora.

Programi iz pozadine

Programi zadnjeg plana su (verovatno već naslućujete) kao stvoreni za podršku periferijskim jedinicama. To su programi koji većinu vremena provode odmarajući se sve dok ih ne prozove program prednjeg plana da reši neki problem vezan recimo za štampač ili disk. Oni se obavezno nalaze u ROM-u i sastavljeni su od više potprograma koji uglavnom deluju nezavisno jedan od drugog. Pre nego što počne da se koristi, pozadinski ROM se mora inicijalizovati. Tu brigu na sebe preuzima takođe program prednjeg plana i ta inicijalizacija se obično vrši na samom početku rada. Prva od rutina u pozadinskom ROM-u se koristi u prethodno navedenu svrhu. Njoj se prosleđuju adrese prvog i poslednjeg slobodnog bejta u memoriji da bi se izjasnila o potrebnom radnom prostoru za valjanu podršku periferijskom uređaju koju će obavljati taj bekgraund program. Ovo se obavlja posredstvom operativnog sistema koji obaveštava forgraund program o tome koliko memorije je pozadinski program rezervisao za sebe. Za svaki program zadnjeg plana koji nam je u radu potreban biće ponovljena ista procedura, a za ovo su predviđene dve rutine operativnog sistema — prva inicijalizuje sve pozadinske ROM-ove, a druga, koja to čini samo za određeni ROM, postoji samo za slučaj kada nam nisu potrebni svi programi ove vrste.

Izgleda da je nastala neopisiva gužva u memoriji našeg jednog „amstrada", sa velikom verovatnoćom da nijedan program ne izvuče živu glavu. Ovo i nije tako daleko od istine. Prilično je teško snaći se sa svom tom količinom programa od kojih svaki zahteva malo radnog prostora. Zbog toga je morala da se primeni dinamička raspodela memorije. Pri tome se i pisci softvera za programe zadnjeg plana moraju pridržavati takve raspodele. To konkretno znači da se nijedan, doslovno nijedan podatak u radnoj memoriji (RAM-u) ne sme adresirati apsolutno. Operativni sistem to podržava tako što se prilikom svakog poziva u neki od pozadinskih programa u indeksnom registru IY nalazi adresa početka radnog prostora tog programa. To je jedini način da se izbegne zbrka i nepredviđena interakcija između raznih programa zadnjeg plana. Programi prednjeg plana nisu opterećeni ovim zahtevima, ali je zato prelazak iz jednog u drugi praćen reinicijalizacijom računara.

Zagonetni RSX

O RSX-u je dosta pisano u svim našim računarskim revijama — verovatno zbog jednostavne upotrebe, pre svega za dodavanje novih bejzik naredbi. Programi RSX-a su inače veoma slični programima zadnjeg plana. Razlika je samo u tome što se nalaze u RAM-u, pa ih je potrebno posle svakog resetovanja računara učitati u memoriju. Pri tom dužnost forgraund programa je da ih predstavi operativnom sistemu, koji ih pridružuje listi spoljašnjih naredbi na kojoj su i naredbe pozadinskog programa. Pri pisanju RSX-a naredbe moraju biti relokatibilne zbog dinamičke raspodele memorije.

Kad god program prednjeg plana ustanovi da je pronašao naredbu koja bi mogla da bude u RSX-u, bekgraund ili forgraund programu, trebalo bi da pozove rutinu operativnog sistema (KL FIND COMMAND) dodajući joj adresu imena naredbe. Operativni sistem se služi listom adresa tabele svih opslužioca spoljašnjih naredbi u koje spadaju samo inicijalizovani RSX i pozadinski programi. Pretraživanje se vrši obrnutim redosledom od redosleda inicijalizacije (poslednja inicijalizovana programska celina se pregleda prva). Ukoliko naredbe nije pronađena, pokušava se sa ROM-ovima programa prednjeg plana počevši od adrese #00. Pri tom je moguća jedna od sledećih akcija:

1) ako je naredbe u nekom RSX-u vraća se bliska adresa rutine

2) ako je naredba u nekom od pozadinskih ROM-ova vraća se daleka adresa koja se sastoji od adrese ROM-a i adrese rutine

3) ako je naredbe u prednjem ROM-u,

Kernal je, kao što i samo ime kaže (kernal-srž), srce jednog operativnog sistema. Za njega su vezane sve bazične operacije koje se odvijaju u jednom računaru. Naš umetak iz prošlog broja „Računara" „Sve amstradove rutine" nije, zbog ograničenog prostora i potrebe za celovitom informacijom, mogao da zađe u objašnjenja osobenosti „amstradovog" operativnog sistema. Ovaj tekst ima upravo takav zadatak i sa objavljenim umetkom predstavlja jedinstvenu celinu.

```
[illegible]
```

dolazi do skoka na traženu rutinu i

4) ako naredba nije pronađena daje se takva indikacija

Svaka od tabela ima sledeći izgled

```
tabela       DEFW lista imena
ulaz — 1       JP rutina — 1
ulaz — 2       JP rutina — 2

lista imena  DEFM  IME   1 '+ #80
             DEFM  IME   .2 + #80

                    DEFB 0  marker
             kraja tabela je obavezan
```

Poslednji karakter u naredbi ima osmi bit setovan i svaki od ulaza mora biti dug tri bajta pri čemu poslednja dva čine adresu rutine

Interapti

Interapt se javlja kao posledica vanrednog stanja do koga dolazi kada je neophodno obaviti neki posao koji nalaže jedan od periferijskih uređaja U tim situacijama procesor prekida dotadašnji posao pamti gde je stao (polaže adresu na stek) i stavlja se na raspolaganje uređaju — počne da izvršava potprogram koji je zadužen za opsluživanje prekida To je ukratko definicija interapta Procesor Z80, oko koga je izgrađen amstrad poseduje više mogućnosti za obradu interapta Mi ćemo opisati samo one mogućnosti koje „amstrad" koristi Procesor se nalazi u interapt modu 1, što znači da svaki prekid izaziva skok na adresu #38 tj izvršenje instrukcije RST #38

U neproširenom računaru postoji samo jedan interapt — vremenski On se javlja 300 puta u sekundi Međutim taj prekid se predstavlja na nekoliko načina

1) Fast Ticker Interrupt (brzi vremenski prekid) — ovaj interapt se koristi za visoku rezoluciju i za precizno merenje vremena (300 puta u sekundi)

2) Sound Generation Interrupt (prekid za generisanje zvuka) — pomoću ovog prekida softverski se generiše envelope ali nije dostupan preostalom delu sistema (100 puta u sekundi)

3) Frame Flyback Interrupt (prekid koji označava završetak generisanja slike) — ovaj prekid se koristi za operacije sa ekranom jer se od njegovog nastanka sledećih par stotina mikrosekundi neće generisati slika pa se mogu izbeći neprijatni propratni efekti na ekranu (50 puta u sekundi)

4) Ticker Interrupt (normalni vremenski prekid) — prilikom ovog prekida „čita" se tastatura (50 puta u sekundi)

Na koji način jedna rutina razlikuje sve ove prekide? Pre svega frejm flajbek je sinhronizovan sa svakim šestim otkucajem brzog časovnika a pri svakom prekidu ispituje se bit 0 porta B čipa 8255 koji je postavljen na 1 dok se slika na ekranu Uveden je i brojač koji se povećava na početku svakog brzog vremenskog prekida i kada dostigne 6 (300/6=50) dolazi do normalnog vremenskog prekida a brojač se resetuje na nulu

Servisna rutina pošto svi zahtevi za prekid „upućuju" na nju mora da razlikuje i spoljašnji prekid od vremenskog To se postiže ponovnim omogućavanjem interapta u samoj rutini pa ako se prekid ponovi znači da se radi o spoljašnjem interaptu

```
rutina
DI
EX AF AF
 u keri flegu uvek se nalazi 0
JR C externa 1
 osim ako je došlo do spoljašnjeg prekida
SCF

EI
EX AF AF
 prekid je omogućen kad prođe jedan mašinski ciklus posle naredbe EI
DI
 ako se radi o spoljašnjem interaptu

doći će do ponovnog prekida

externai
POP HL
 jedna adresa je višak
AND A
 „resetuj keri"
EX AF AF
 i vrati ga nazad
CALL #3B
 pozovi rutinu za obradu spoljnih prekida
```

Prethodno mora da se instališe rutina koja servisira spoljašnje prekide Ako je

zaključeno da je u pitanju prekid koji je generisala periferna jedinica poziva se potprogram na adresi #3B Na to mesto se prilikom ubacivanja rutine postavlja 5 bajtova koji će, verovatno uključivati i neku JP instrukciju pri čemu se prethodni sadržaj tog dela kopira na neko drugo mesto Ova procedura je potrebna iz prostog razloga što je moguće da je prethodno bila instalirana neka rutina koja je intervenisala pri prekidima sa drugog perifernog uređaja Ako rutina za servisiranje zaključi da nije kompetentna za obradu prekida koji je nastao trebalo bi da skoči na kopiju tih 5 bajtova Jasno zbog mogućeg prebacivanja taj deo dužine 5 mora biti nezavisan od adrese na kojoj se nalazi tj relokatibilan Poslednja kopija kojih može biti neograničeno mnogo jeste ona koju sistem postavlja pri inicijalizaciji a to je jednostavno, povratak — naredba RET Rutina za servisiranje ima zadatak da bar resetuje izvor interapta da ne bi došlo do beskonačne petlje

i događaji

Sve što je do sada iznelo u ovom tekstu već je manje-više viđeno Ovo što sada dolazi predstavlja zaista novost u projektovanju operativnih sistema za mikroračunare

Događaji (events) su uvedeni prvenstveno zbog potrebe da se osetljivo procesiranje spoljašnjih prekida obavi van rutine za obradu interapta Zbog specifičnog načina obrade interapta rutine za procesovanje spoljašnjih prekida su opterećene nizom zahteva moraju biti kratke ne smeju da omoguće prekid i moraju se nalaziti u RAM-u koji je već opterećen radnim prostorom OS-a i programe prednjeg i zadnjeg plana Programu za obradu prekida je omogućeno da hardversku pojavu — interapt — dalje transformiše u softversku pojavu — događaj

Tehnika je sledeće kada rutina za obradu spoljašnjih prekida ustanovi da može da obradi nastali prekid ne obrađuje ga već samo resetuje izvor prekida i zahteva od operativnog sistema da priključi na jedan od redova čekanja (pending queues) događaj Od prirode prekida zavisi kada će se on obraditi i o tome odlučuje rutina koja je poslala zahtev za inicijalizovanje događaja

Pronaći ćete na mnogim mestima (pa čak i u oficijelnom Firmware manual-u) da je događaj softverski ekvivalent interapta To je donekle tačno jer je svaki interapt moguće prevesti u događaj ali nisu svi događaji dobijeni na taj način To znači da postoje događaji koji su nezavisni od interapta ako ga posmatramo kao postavljanje INT nožice mikroprocesora u aktivno stanje U principu moguće je definisati bilo kakve uslove pod kojima će se javljati zahtevi za obradu događaja

Događaj se sistemu predstavlja kao blok podataka Blok se sastoji od 7 bajtova

0 1 Chain — ovaj podatak postavlja i koristi samo operativni sistem da bi mogao da ulanča listu blokova

2 Count — ovaj bajt se koristi kao brojač razlike između broja zahteva za obradom događaja i broja obrađenih zahteva (posebno značajno za sinhrone događaje)

3 Class — klasa događaja

Bit 0 1 — ‚bliska' adresa rutine
0 — ‚daleka' adresa rutine
Bit 1—4 prioritet sinhronih događaja
Bit 5 0
Bit 6 1 — ekspresni događaj
0 — normalni događaj
Bit 7 1 — asinhroni (neusklađeni) događaj
0 — sinhroni (usklađeni) događaj

4,5 Address — adresa rutine za obradu događaja

6 ROM — adresa ROM-a u kom se nalazi rutina za obradu događaja (ukoliko je u klasi naznačeno da se rutina nalazi na ‚bliskoj' adresi — bit 0 setovan— ovaj podatak se ne uzima u obzir)

Brojač (Count) se prilikom inicijalizacije bloka postavlja na 0 Kasnije se prilikom svakog zahteva povećava i posle svake obrade događaja smanjuje za 1 Da nema ovog podatka Kernel bi ispustio zahteve ako ih se više javi pre obrade događaja Brojač se može upotrebiti i za onemogućavanje događaja ako ga postavimo na negativnu vrednost ili za odbacivanje neželjenih zahteva postavljanjem na 1

Klasu (Class) događaja postavlja rutina koja ga inicijalizuje Ona određuje da li će događaj biti sinhroni ili asinhroni ekspresni ili normalni i okvirno položaj rutine za obradu događaja „Bliska' adresa označava da se rutina nalazi u donjem ROM-u ili centralnom delu RAM-a a ‚daleka' da se nalazi u nekom od bočnih ROM-ova Pri tome ekspresni asinhroni događaji moraju biti na ‚bliskoj' adresi Prioritet sinhronih događaja može biti od 0 do 15 (što je veća vrednost veći je i prioritet)

Iza svakog bloka događaja mogu se nalaziti korisnička polja (user fields) koja će koristiti rutina za obradu ili (i) neke druge programske celine Prilikom svakog poziva rutine za obradu događaja kernel prosleđuje u nekom od registara adresu preko koje se može izračunati adresa polja (BLK je adresa početka bloka)

— ako je rutina na ‚bliskoj' adresi registar DE će sadržati BLK+5 (u knjizi ‚Firmware manual' ovde je data netačna informacija da je u HL BLK+6)

— ako je rutina na ‚dalekoj' adresi registar HL će sadržati BLK+5

Tako je pružena mogućnost da se samo jedna rutina za obradu događaja koristi za obradu sličnih događaja, ukoliko se pridržava pravila relativnog adresovanja podataka (setite se dinamička raspodela memorije)

Zahtev i obrada

Moguće je napraviti jedno zanimljivo iako grubo poređenje između raznih vrsta događaja i ponašanja čoveka u nekim situacijama Zamislite da ste uzeli da sređujete svoj album sa značkicama iznervirali ste se što vam je album u lošem stanju, koncentracija je popustila i uboli ste se Odmah ste trgli ruku uz nekoliko prigodnih opaski Naravno to bi bio jedan ekspresni asinhroni događaj doduše propraćen nekim uzgrednim manifestacijama u čiju se prirodu sada nećemo upuštati jer se nikad ne dešavaju u računaru (ukoliko neko ipak zna jedan takav primer bićemo vrlo srećni da ga objavimo) Zamislite samo kako tek razne pentanje bockaju računar Došla je zima uskoro će početi i već tradicionalna restrikcija struje Ustanovili ste da je ostalo još pola časa do nominovnog isključenja računara i počinjete da tražite kasetu na koju će biti snimljeno novo remek-delo programiranja Snimili ste program i najzad krenuli na spavanje To je otprilike jedan normalni asinhroni događaj osim ako vas je nestanak struje zatekao nespremnog ali tada će doći samo do već pomenutih uzgrednih manifestacija (rekli smo da to ipak nije predmet ovog izučavanja) A šta ćete uraditi ako se na vratima vaše sobe pojavi neko od bližnjih ukućana i opomene vas da bi možda bilo krajnje vreme da već jednom sredite svoju sobu? Naravno, stavićete tu primedbu na listu čekanja u stvari pošto je već na listi čekanja, samo ćete povećati brojač za jedan i nastaviti sa svojom igricom Ukoliko imate posebno pedantne roditelje verovatno će vam biti potreban brojač od 16 a ne od 8 bita To je jedan sinhroni događaj i to vrlo niskog prioriteta

Autor ovog teksta se čuvao greške koju je lako načiniti ako se mehanizam za procesiranje događaja ne shvati dobro Na većini mesta se pominje reč zahtev, a ne obrada Programer vodi računa o zahtevima za obradu događaja a kernel o samoj obradi Granica između dva pomenuta pojma je strogo postavljena i oni se razmatraju sasvim odvojeno

Dakle, postoje dve osnovne klase događaja

a) sinhroni događaji — upotrebljavaju se za prekide kod kojih je potrebna gotovo trenutna akcija Mogu biti

1) ekspresni — obrada događaja se vrši odmah — iz rutine za obradu prekida

2) normalni — blok koji opisuje događaj smešta se u red čekanja i obavlja neposredno pre izlaska iz rutine za obradu prekida

b) sinhroni događaji — koriste se za interakciju među raznim delovima glavnog programa Takođe mogu biti ekspresni i normalni O tome kada će biti procesovani, odlučuje glavni program

Shodno prethodnoj podeli, postoje i dva reda čekanja koja se sasvim odvojeno procesuju asinhroni i sinhroni red čekanja Blokovi koji opisuju događaje ulančani su u liste pomoću sistemskog podatka (Chain) koji se nalazi na početku svakog bloka

Asinhroni događaji

Ekspresni asinhroni događaji se koriste samo kad je potrebna zaista hitna akcija Rutina za njihovu obradu je opterećena istim zahtevima kao i rutine za obradu samih prekida

Normalni asinhroni događaji su najlakši za upotrebu Brzina kojom se pozivaju uglavnom zadovoljava sve zahteve koje postavljaju periferne jedinice Ako se zahtev javi van interapta (od strane glavnog programa) događaj se procesira odmah Rutina za obradu ovih događaja koriste interni stek dužine 128 bajta Mogu trajati koliko je potrebno a svi zahtevi nastali u međuvremenu biće uzeti u obzir i opsluženi po završetku obrade zahteva koji je prispeo

Asinhroni događaji mogu se onemogućiti slično maskiranom interaptu To može da se učini postavljanjem brojača na negativnu vrednost ili pozivanjem za to predviđene rutine (KL DISARM EVENT) koja radi istu stvar Kasnije se mogu omogućiti inicijalizovanjem bloka (KL INIT EVENT) ili postavljanjem brojača na nulu Onemogućavanje događaja ne sprečava kernel da po svakom zahtevu za obradu pogleda da li je

događaj omogućen ili ne. Ako bi se zahtevi slali često došlo bi do priličnog usporanja čitavog sistema. Zato je bolje svaki onemogućeni asinhroni događaj skinuti sa njegove liste čekanja.

Sinhroni događaji

Sinhroni događaji se nalaze pod kompletnom kontrolom glavnog programa. Kada se javi zahtev za obradom ove vrste događaja, blok događaja se postavlja na jedan od redova čekanja ali se ne obrađuje sve dok forgraund program to ne naredi. Zato bi trebalo da program prednjeg plana kada god „uoči" da sinhroni red čekanja nije prazan pozove kernelove rutine za obradu sinhronih događaja, mada u krajnjoj liniji nije obavezan da obavi tu proceduru. Takođe postoje normalni i ekspresni sinhroni događaji, ekspresni događaji imaju viši prioritet od normalnih.

Sinhrone događaje korisnik može onemogućiti korišćenjem rutine KL DEL SYNCHRONOUS koja istovremeno skida i blok koji opisuje događaj sa liste. Postoji i rutina (KL DISARM EVENT) koja onemogućava sve normalne sinhrone događaje, dok se ekspresni normalno obrađuju.

Ima nekoliko primera upotrebe sinhronih događaja u neproširenoj mašini. Tu su naravno naredbe bejzika za „rad se interaptom" (razume se, ne radi se sa spoljašnjim već sa unutrašnjim vremenskim interaptom) koje su jedan od glavnih „amstradovih" aduta. Atraktivne naredbe nema šta, ali za njih je ponajmanje zaslužan bejzik. Čitav posao obavlja kernel, a radi se o normalnim sinhronim događajima. Naime, kada interpreter naiđe na naredbu EVERY ili AFTER, inicijalizuje jedan normalni sinhroni događaj, pri čemu će rutina koja opslužuje događaj biti sam interpreter. Pri svakoj obradi zahteva izvršiće se potprogrami koji počinju od programske linije navedene iza svake od dve pomenute naredbe. Posle svake naredbe koja se interpretira, bejzik proverava da li je sinhroni red čekanja prazan i ako nije, obrađuju se svi zahtevi koji su pristigli.

Drugi primer je događaj koji generiše operativni sistem prilikom čitanja tastature. Kad god uoči da je taster ESCAPE pritisnut, šalje zahtev da se obradi jedan ekspresni sinhroni događaj koji obrađuje zahtev korisnika da se prekine program. U većini slučajeva rutina za obradu ovog događaja samo smešta marker u karakter-bafer koji obaveštava program koji se izvršava (recimo bejzik interpreter) da je korisnik uputio zahtev za obradu „brejka".

Svaki od ova dva primera ima neke veze sa interaptom — prvi je direktna posledica normalnog vremenskog prekida, dok drugi nastaje sa tastature koja se skanira pri istom prekidu. Navešćemo sada jedan primer gde se obrada događaja odvija sasvim nezavisno od bilo kakve hardverske pojave. Radi se o naredbi čiju upotrebu omogućava deo operativnog sistema koji se bavi kontrolom zvučnog čipa i generisanjem zvuka. Zvučni čip AY-3-8912 je, da dozvolimo sebi malo literarne slobode, tročlani orkestar. To znači da ima tri kanala i svaki od njih može da ispušta nekakav zvuk, pa ako ste dobar dirigent, to može da liči na muziku. Svakom od kanala je dodeljen po jedan red čekanja u koje se može smestiti najmanje po tri zvuka, tj. zvučne naredbe koje sasvim određuju kako će izgledati ono što neki od kanala ispusti. U bejziku postoji naredba koja ima sledeći izgled

ON SQ(kanal) GOSUB nnnn

One se interpretira na sledeći način: definiše se i inicijalizuje blok koji opisuje jedan događaj (tj. potprogram koji počinje od linije nnnn) i šalje obaveštenje kontroloru zvuka gde se nalazi blok i za koji kanal je događaj zadužen. Zahtev za obradu događaja će biti poslat kada se nađe prazno mesto u redu čekanja koji je pridružen tom kanalu. Najverovatnije je da bejzik interpreter definiše sinhroni događaj (pogodite za-što!), dok bi mašinski program radije defi- sao neki normalan asinhroni događaj. O... naredba uveliko pojednostavljuje pravljen... muzike. Kada god postoji mogućnost da ... nekom od redova doda zvuk, biće pozv... potprogram čiji je zadatak da pogleda k... je nota na redu i da je doda redu. Tako ... se odvija automatski i transparentno ... stanovišta glavnog programa koji za ... vreme može nesmetano da obavlja s... poslove.

Tri izvora

Pored već pomenutih ... nal ima definisana tri izvo... događaja i prema toma tri gr...

1) Fast Ticker Events — 3... sekundi tj. kada se javi brzi vr... interapt

2) Frame Flyback Events — na svakog generisanja slike (50 puta u sekundi)

3) Ticker Events — pri normalnom vremenskom prekidu (50 puta u sekundi) generišu se zahtevi za obradu ovih događaja — koriste se od strane bejzika za interpretaciju EVERY i AFTER naredbi.

Blokovi ovih grupa su nešto prošireni, ali na kraju svakog od njih nalazi se blok događaja koji smo već opisali (dužina 7 bajta).

Prve dve grupe imaju opisne blokove dužine 9 bajta, pri čemu su prva dva bajta adresa (Frame Chain ili Fast Chain) koju koristi kernel za ulančavanje odvojenih lista svake od grupa.

Treća grupa događaja je izuzetno fleksibilna. Blok koji ih opisuje je dugačak 13 bajta i ima sledeći izgled:

0 1 Tick Chain — adresa koja služi za ulančavanje blokova

2 3 Tick Count — brojač otkucaja

4,5 Recharge Count — vrednost na koju se postavlja prethodni podatak posle svakog zahteva za obradom događaja

6+ blok događaja

Zahtev za servisiranjem ovih događaja se ne javlja posle svakog vremenskog prekida, već se samo brojač otkucaja smanjuje za jedan. Kada dostigne nulu, šalje se zahtev, a brojač se postavlja na vrednost koju definiše Recharge Count.

... sve ... sor ... ND ... XT ... sli- ... Novi ... ovanja je ... roni događaj ... normalnih vremenskih prekida. Rutina može da kontroliše samo jedan kursor. Za drugi (COPY) kursor potrebno je izmeniti odgovarajuće ulaze u glavnom džamp-bloku (TXT PLACE CURSOR i TXT REMOVE CURSOR), kao i glavne rutine. Rutina koja servisira događaj može ostati ista jer se pridržava relativnog adresiranja. Program je relativno dugačak pre svega zbog loše projektovanog linka. Nova rutina koja zamenjuje osnovnu ne dobija nikakve informacije od strane sistema, mora sama da pronađe gde treba smestiti kursor, da li je potrebno skrolovati prozor, pa čak i da ustanovi da li je kursor omogućen ili ne, za šta ne postoji način da se utvrdi posredstvom džamp-bloka. Zbog ovoga, autor nije siguran da li će program raditi na CPC664 i CPC6128 modelima. Ipak, učinjeno je sve da ne dođe do kraha i verovatno je ipak ostvarena kompatibilnost.

Dejan Muhamedagić

podmlađeni BBC

Naš test
Drugi 6502

Sa donje strane BBC-ja su smešteni portovi za priključivanje raznih dodataka — disk jedinice štampača modema kao i naizbežni port opšte namene Sa krajnje desne strane računara je smeštan priključak nazven *Tube* pri čemu se pri svakom pominjanju ovog porta kaže da je *Tube* registrovani trademark Acon-a BBC naime hardverski i softvarski podržava povezivanja se spoljnim mikrokompjuterima koji će preuzeti sve njegove vitalne funkcije svodeći bazični računar na inteligentni grafički terminal Acorn tranutno prodaje osmobitne koprocesore 65C02 i Z80B kao i šesnaestobitni koprocesor 32016, cene ova tri uređaja iznose respektivno 200 400 i 1400 funti

S kraja ka početku

Podimo suprotno uobičajanom poretku stvari od kraja i pogledajmo fascinantne karakteristike koprocesora 32016 National Semiconductor 32016 je trideset dvobitni mikroprocesor kod nas prilično zasenjen Motorolom 68000 32016 je međutim procesor u svakom pogledu superioran Motorolinom čipu i pripada klasi mikroprocesora 68020 od koga je noviji i prema tome bolji Za sada se uglavnom primenjuje na velikim kompjuterima firme Digital Research i na nekim IBM-ovim modelima, dok mu je BBC prvi izlazak u svet mikroračunara Acorn ga je u startu opremio jednim megabajtom RAM-a (adresni prostor od 64 megabajta verovatno neće biti popunjen u skoroj budućnosti) operativnim sistemom PANOS (verzija unix-a, pravi Unix može da se primeni tek kupovinom hard diska) i kompajlerima za Fortran 77 pascal C kao i BBC bejzik interpretatorom Mogu se nabaviti i kompajleri i interpretatori drugih programskih jezika kao što su Lisp Logo i prolog Acorn planira da 32016 (i njihov tranutno vodeći računar Cambridge Workstation) bude kompatibilan (i po brzini rada superioran) sa u svetu i kod nas sve popularnijim mini računarima VAX koji, premda obično rade pod operativnim sistemom VMS mogu da rade i pod Unix-om Tome pomaže i aritmetički koprocesor NS32081 koji obavlja množenja, deljenja i druge operacije sa brojevima u pokretnom zarezu mnogo puta brže nego što stari Z80 i 6502 sabiraju intidžere! Čini nam se da se nekome ko planira da nabavi Cambridge Workstation tranutno više isplati da kupi BBC hard disk i 32016, jeste da će dati stotinak funti više ali će bar imati i BBC koji bez svih ovih skupih dodataka može da izvršava razne „smrtne" programe koji uključuju i dobre igre

Koprocesor Z80B se isporučuje sa 64 K RAM-a (bedno zvuči prema onim megabajtima ali) Z80B je verovatno, pripremljen da bi učinio BBC kompatibilnim sa poznatim operativnim sistemom CP/M pa ćete tako uz sam uređaj dobiti i četiri diskete pune CP/M softvera teksl procesor baza podataka unakrsna izračunavanja, makro bejzik Acorn je verovatno u pravu kada kaže da sav ovaj softver vredi na slobodnom tržištu više od 400 funti koliko Z80B košta Sa druge strane verovatno nećete biti zadovoljni svim ovim programima pa ćete postepeno kupovati skuplje (ako mislite da ćete ih lako i besplatno pokupiti Iz neke firme koja koristi Partnera varate se — različiti formati diskova će vas dobro namučiti) Najneprijatnija osobina Z80 koprocesora je što ni jedan ROM kojim ste uz dosta izdataka opremili vaš BBC neće moći da koristi njegove potencijale — program pisan za 6502 *nikako* ne može da radi na Z80! Ukoliko ste se dakle, navikli na teksl procesor View ili na bazu podataka View Store moraćete da se od njih odviknete! Zbog svega toga Z80B predstavlja prilično promašen koprocesor koga slabo koji pojedinac nabavlja — odgovara jedino firmama koje žele da unificiraju operativne sisteme na raznim računarima Vlasnici BBC-ja pa tako i autor ovoga teksta, se obično odlučuju za nabavku koprocesora 65C02 koji je na neki način najbolje primeran BBC-ju Ni parametar da se radi o najjeftinijem nabavljivom koprocesoru (svega 200 funti) svakako nije bez značaja

U visokom društvu

Procesor 65C02 je upakovan u kutiju veliku koliko i pola BBC-ja postavlja se uz sam računar se kojim je spojen neprijatno kratkim kablom Sa zadnje strane uređaja je uz naizbežni osigurač, smešten prekidač pomoću koga ga uključujete ili isključujete kada isključite 65C02 i pritisnete BREAK radićete sa sasvim običnim ' BBC-jem i tako rešiti sve probleme kompatibilnosti o kojima ćemo govoriti tek na kraju ovog prikaza

Nije međutim dovoljno samo da priključite 65C02 i počnete sa radom najpre je potrebno otvoriti BBC-jevu kutiju i zameniti DFS ROM (ROM u koji je upisan operativni sistem za rad se diskovima) novim DFS NET ROM-om! U ovaj je čip osim unapređenih rutina za komunikaciju sa diskom i računarskom mrežom (Econet) smešten i softver za komunikaciju sa koprocesorom (osnovni operativni sistem poziva ove module preko takozvanih „proširenih vektora', o njima nekom drugom prilikom) Ako ste se, međutim, navikli na korišćenje nekog drugog DFS-a i njegove karakteristike kao što su dvostruka gustina upisa ili više od 32 datoteke na disketi, povratak na Acornov DFS će vam biti prava beda Rešenje, na sreću, postoji ostavite stari DFS u računaru a zatim postavite DFS NET tako da bude u podnožju nižeg prioriteta (levo od DFS-a koji koristite) Nevolja kod ovoga rešenja je što će ako isključite koprocesor PAGE biti na & 1D00, što znači da ćete u osnovni BBC moći da učitate samo najkraće programe Watford Electronics DFS 1 43 je uspešno rešio i ovaj problem ovaj ROM na neki volšeban način „ubija" deo DFS NET-a tako da možete da koristite kako osnovni BBC (PAGE na & 1900) tako i dodatni procesor

Kada najzad završite sa povezivanjem, uključite oba računara i pritisnite CTRL BREAK, na ekranu će se umesto standardnog „BBC Computer 32 K' pojaviti napis „Acorn Tube 64K' — u kutiju sa dodatnim procesorom je naime, ugrađeno 64 K dinamičkog RAM-a koji bi praktično u celini trebao da bude slobodan za programe i podatke Pokušajmo da se uverimo u to otkucavši PRINT (HIMEN-PAGE) 1024 na ekranu će se pojaviti broj 30 što znači da je za bejzik slobodno svega 30 kilobajta U čemu je problem?

Na slici 1 je prikazan adresni prostor 65C02 koprocesora Početak i kraj adresna

BBC B je poznat kao odličan i nadasve brz računar sa brojnim proširenjima i dobrom softverskom podrškom, ali i kao mašina sa malom memorijom — svega 32 K. Trideset dva kilobajta i ne bi bila toliko nedovoljna kada BBC ne bi imao odličnu grafiku — u najvišem grafičkom modu ekran zauzima čak 20 K RAM-a što znači da će za bejzik programe preostati 8.5 kilobajta (svega 5.75 K ako koristite disk) — premalo da bi se bilo šta ozbiljno preduzelo. BBC-jev RAM se, naravno, može proširivati bilo kupovinom raznih RAM tabli bilo zamenom za model B+ kod koga je video memorija odvojena od osnovnog adresnog prostora. Pa ipak, pravo proširenje zamišljeno u doba koncipiranja ovoga računara je dodatni procesor.

slika 1

Oblast memorije	Adresa
	FFFF
Tube FIFO	
OS dodatnog 65C02	F800
Slobodno	C000
BASIC ili neki drugi jezik	8000
BASIC stek i promenljive	HIMEN LOMEN
BASIC program	0800
Radni prostor jezičkog ROM-a	PAGE 0400
Sistemske promenljive i stek	0000

mape su rezervisani za stek sistemske potrebe i FIFO stek preko koga se na kloku od 2 MHz komunicira sa osnovnim procesorom. BASIC ROM zauzima adrese od &8000 — &C000 — jednostavno je prekopiran u adresni prostor koprocesora. Prostor između &C000 i &F800 je potpuno slobodan — u osnovnom modelu računara ovaj prostor zauzima operativni sistem koji dodatnom procesoru nije potreban pošto se u potpunosti oslanja na I/0 processor. Tih 14 K možete da iskoristite za podatke tako što ćete na početku bajzik programa napisati LOMEN=&C000. Acorn se međutim setio i načina da se ovaj prostor bolje iskoristi uz dodatni procesor ćete primetiti i ROM sa takozvanim HI-BASIC-om.

Ukoliko planirate da koristite HI-BASIC moraćete još jednom da rasklopite BBC i ugradite ovaj čip u jedno od podnožje (ako nemate više slobodnih podnožja, krajnje je vreme da kupite neku ROM tablu) tako da se nalazi levo od osnovnog BASIC ROM-a. Kada docnije uključite čitav sistem otkucaćete *FX 142 n gde ste sa n označili broj podnožja u koje ste ugradili HI-BASIC. PRINT (HIMEN-PAGE) 1024 će vas uveriti

slika 2

Oblast memorije	Adresa
	FFFF
Tube FIFO	
OS dodatnog 65C02	F800
HI-BASIC ili drugi HI-ROM	B800
BASIC program i podaci	HIMEN 0800
Radni prostor jezičkog ROM-a	PAGE 0400
Sistemske promenljive i stek	0000

da vam je za bejzik programe i podatke sada raspoloživo maksimalnih 44 kilobajta.

Slika 2 ilustruje rad HI-BASIC-a. radi se o bejzik interpretatoru koji je asembliran tako da se upisuje u memoriju počevši od &B800 (a ne od &8000) što znači da ga sa *SAVE HIBASIC 8000 +4000 možete snimiti na disketu, odstraniti ROM sa njim iz računara i ubuduće ga startovati sa *HIBASIC. Acorn je najavio da će prodavati 'HI' verzije ostalih uslužnih ROM-ova koje je izbacio na tržište — tek će tako 6502 koprocesor biti u potpunosti iskorišćen. Za sada je u prodaji HI-VIEW (tekst procesor koji oslobađa 44 kilobajta za tekst) i u trenutku kada ovo budete čitali, HI-COMAL.

Kompatibilno ili ne

Mnoga nezavisna softverska firme nisu na žalost spremne da proizvode HI verzije svojih programa. Da bi program radio u memoriji koprocesora on mora da bude pisan tako da koristi *jedino* dokumentovane rutine operativnog sistema. Pogledajmo jednostavan primer: program koji se izvršava je smešten u adresni prostor koprocesora dok je video memorija smeštena u adresni prostor I/0 procesora. Ukoliko program koji se izvršava crta po ekranu pokušavajući direktno u video memoriju, na ekranu se ništa neće videti!

Praktično ni jedna komercijalna igra ne radi dok je dodatni procesor uključen, što je prilično razumljivo i nije zabrinjavajuće. Kažemo da je nekompatibilnost igara razumljiva pošto se kod njih, zbog zahteva za brzinom, obično direktno pristupa video memoriji. Kažemo da njihova nekompatibilnost nije zabrinjavajuća jer igre rade sasvim lepo kada je koprocesor isključen i verovatno bi kada bi ih on izvršavao bile prebrze i neupotrebljive. U poslednje vreme se pojavljuju igre pisane specijalno za BBC sa dodatnim 65C02 programa kao što je elite 2 bi svakako trebalo nabaviti!

Ako o (ne)kompatibilnosti igara ne vredi mnogo razmišljati, nekompatibilnost svakog poslovnog i korisničkog programa predstavlja pravu malu tragediju, pogotovu kada se radi o dobrom ROM-u koji bi itekako mogao da iskoristi veću memoriju i brži rad dodatnog 65C02. Svi ROM-ovi koji sadrže samo servisne rutine (PRINTMASTER, DUMPOUT, Graphic ROM, Disc Doctor...) su 100% kompatibilni sa dodatnim 65C02, jer ovaj nema nikakve veze sa njihovim izvršavanjem. Programski jezici (paskal, komal, forth i razni asembleri (npr ade) su takođe kompatibilni sa dodatnim procesorom, dok se najveći problemi javljaju sa pojedinim poslovnim programima. Wordwise je, na primer, nekompatibilan, a Wordwise Plus samo delimično kompatibilan (editovanje i dalje mora da se obavlja u modu 7, tj. bez naših slova) sa koprocesorom. Inter Chart i Inter Sheet rade savršeno. Starbase ne radi zbog zaštite (verovatni ili je nesreća što Beebugsoftovi zaista izvanredni ROM-ovi Toolkit (skup komandi koja pomažu bejzik programerima), Exmon 2 (monitor / disasembler) i Slauth (ROM koji omogućava izvršavanje bejzik programa naredbu po naredbu) nisu kompatibilni sa koprocesorom, što za čoveka koji se navikao na njihove usluge predstavlja veliku neprijatnost. Exmon 2 se mora zameniti daleko slabijim Gremlin-om (100% kompatibilan), dok je Toolkit jednostavno nezamenljiv. Treba sačekati da iz Engleske pristigne već naručeni Toolkit Plus i videti da li će on radiji sa dodatnim 65C02.

Najinteresantniji su ROM-ovi sa programima koji se samostalno prilagođavaju osnovnom ili dodatnom procesoru. Ovi ROM-ovi će neke svoje opcije primenjivati samo u radu sa dodatnim 65C02, dok će vlasnici osnovnog BBC-ja moći da ih koriste u zadovoljavajućoj meri. Jedine dva programa ovoga tipa koja su nam došla do ruku su Technomatic-ov novacad (Computer Aded Design) i View 3.0.

I/O procesor

Verovatno ste se već zapitali a šta radi stari BBC kada je dodatni procesor preuzeo sve poslove? Stari BBC je sveden na takozvani I/O procesor: brine se o tastaturi, ekranu, štampaču, disku, računarskoj mreži, prima i obrađuje interapte od drugih uređaja i uopšte, brine o periferiji. Dodatni procesor će tako poslati koordinate tačaka koje treba spojiti linijom a zatim nastaviti da izvršava program dok će I/O procesor raditi na crtanju te linije. Ukoliko su dakle, programi dobro pisani (malo računanja pa malo crtanja i tako dalje), biće ostvaren paralelizam u radu, što znači da će se programi u kombinaciji BBC/dodatni 6502 izvršavati čak dva puta brže od programa na osnovnom modelu računara. Razliku u brzini koja je posledica kako paralelizma u radu tako i bržeg clock-a pokazuju čak i jednostavni Benchmark testovi koje smo objavili u „Računarima 9".

Memorija I/O procesora se koristi za pamćenje ekrana, definicije tastera i predefinisanih karaktera (možete da predefinišete *sve* znakove a da vas to ne košta ni jednog bajta memorije u koju se upisuje bajzik programi), kao radni prostor aktivnih i neaktivnih ROM-ova i za mnoge druge sistemske potrebe. Ukoliko dobro poznajete asembler, možete čak i da pišete programe koji će paralelno izvršavati na dva procesora!

Sve u svemu, dodatni 65C02 predstavlja proširenje primereno BBC-ju koje obezbeđuje komforan i brz rad i kompatibilnost sa većinom već nabavljenih ROM-ova. Cena mu je naravno previsoka: za 200 funti bi se mogao kupiti QL, Amstrad ili neka mašina iz slične klase. Ipak, za ljude koji planiraju da još izvesno vreme ostanu na BBC-ju dodatni 65C02 predstavlja odlično rešenje.

Dejan Ristanović

Na drugi način

Poker bez karata

Svako ko je ikada igrao karte zna da se događa da poneko, tu i tamo, pokušava (pa i uspeva) da vara. A kakva je tek prilika da se vara kad se poker igra bez karata, recimo telefonom. Da li biste se zaista usudili da igrate poker, a da karte deli vaš protivnik i o tome vas obaveštava telefonom? „Da li je moguće igrati poker bez karata, a da svi učesnici igre budu u ravnopravnom položaju, da niko od njih ne može da vara, a da ostali to uvek, kad neko pokuša, odmah primete?" — upitao se jednom Robert Flojd (Robert Floyd) sa Stanford univerziteta u Kaliforniji, inače dobitnik Tjuringove nagrade. Odgovor na ovo pitanje dali su generisanjem potrebnog algoritma Adi Samir (Adi Shamir), Ronald Rajvest (Ronald Rivest) i Leonard Adelman sa MITa u Bostonu.

Igra počinje poštenom deobom karata što podrazumeva da je svaki igrač dobio karte na potpuno slučajan način — da niko sem njega ne zna karte koje je dobio i da ne postoji karta koju su u toku partije dobila dva igrača. Naravno svaki igrač mora da zna koje je karte dobio a o protivničkim ništa više sem očigledne informacije da protivnik nema ni jednu od karata koju ovaj ima. U toku igre može doći do situacije da neki od igrača uzme kartu sa vrha špila i algoritam mora da obezbedi da igrač može da uzme potrebnu kartu a da pri tom ni o njoj ni o nekoj drugoj karti koja je kod njega protivnicima ne da nikakvu informaciju. (U nekim drugim igrama sa kartama pojavljuje se situacija da jedan igrač treba da dâ drugom neku kartu tako da je niko od ostalih učesnika u igri ne vidi pa algoritam mora i ovakvu mogućnost da realizuje.) Na kraju svake partije svi igrači moraju imati mogućnost da provere da li je igra bila poštena ili je neko pokušavao da vara. Recimo, ako je neko u toku partije izjavio da ima četiri kralja, svi igrači mogu da se na kraju partije uvere da ih je on stvarno i imao (ili u nekim drugim igrama da nije imao određenu kartu). Na prvi pogled nije baš jednostavno realizovati traženi algoritam, ali ako svaki od igrača ima pri ruci i kompjuter i odgovarajuću količinu domišljatosti ili znanja, a najbolje i jedno i drugo, algoritam se može realizovati. Pre nego što pročitate članak do kraja, pokušajte da rešite problem — javite nam ako uspete na neki drugi način.

Pretpostavimo da poker igraju dvoje, Ana i Mirko, i da se pre početka igre dogovore o dve funkcije K (kodirajuća funkcija) i D (dekodirajuća funkcija) sa sledećim osobinama:

1 Kc(X) kodirana varijanta signala X u kodu c (signalom X igrač javlja partneru da je dobio kartu X)

2 Dc(Kc(X))=X pri bilo kom X ili pri bilo kom kodu c

3 Kc(Kd(X))=Kd(Kc(X)) za bilo koju kartu X i kodova c i d

4 ako su X i Kc(X) zadati ni najbolji razbijač šifri ne može otkriti kod c za bilo koje X i c

5 za bilo koja dva signala X i Y nemoguće je izabrati kodova c i d tako da je Kc(X)=Kd(Y)

Osobina 3 je malo neobična ali je ostvariva, osobina 4 označava kriptografsku sigurnost, a osobina 5 služi za onemogućavanje varanja kada se na kraju partije vrši provera da li su svi igrači igrali korektno. O tome kako se generišu ovakve funkcije možete pročitati u sledećem broju „Računara" u članku o kriptografskim protokolima, ako u međuvremenu ne smislite takve funkcije. Dogovorivši se o funkcijama K i D Mirko i Ana tajno jedno od drugog biraju kodove M i A. Oni ove kodove čuvaju do kraja igre kada ih pokazuju jedno drugom da bi se uverili u poštenu igru. Mirko generiše 52 signala (pravi poker se igra sa 52 karte), recimo brojeve od 1 do 52, i kodira ih kodom M i zatim ih u slučajnom poretku šalje Ani. Slučajni poredak ostvaruje tako što promeša karte pa ih po poretku špila pošalje Ani ili još bolje i brže pomoću generatora slučajnih brojeva u svom računaru. Ana bira 5 signala (karti) i šalje ih Mirku, koji ih dekoduje. Pošto je kod M poznat samo Mirku, Ana ništa ne zna o Mirkovim kartama do kraja igre. Zatim Ana šalje još 5 signala Mirku ali ih prethodno koduje svojim kodom A. Svaki od signala je dvostruko kodovan K (K (n)) što je po osobini 3 jednako K(K (n)) za bilo koje n. Mirko dekoduje signale i dobija K (n) i šalje ih nazad Ani, koja dekoduje signal i dobija svoje karte koje Mirko nije mogao da vidi pošto ne zna kod A.

Majkl Rain sa Harvarda u Bostonu predložio je zanimljivu fizičku realizaciju algoritma. Kodovanje predstavlja kao stavljanje katanca na kutiju sa kartom. Mirko prvo stavlja svaku kartu u posebnu kutiju i na svaku kutiju stavlja katanac za koje ima isti ključ M. Ana bira 5 kutija i šalje ih Mirku koji ih otključava i vidi svoje karte. Zatim Ana stavlja na 5 slučajno izabranih kutija i svoj katanac, kutije šalje Mirku koji otključava svoj katanac, šalje kutije Ani i pošto ih ona otključa vidi svoje karte koje Mirko nije mogao da vidi. Analogija sa izmenjenim redom otključavanja katanaca je potpuna kao u algoritmu sa kodovanjem. Ako u toku igre neko od igrača traže karte, algoritam je isti kao kod podele karata. Na kraju igre igrači otkrivaju svoje kodove i uveravaju se da partner nije varao. Ova procedura se očigledno lako realizuje i kad je neka druga igra kartama u pitanju ili kada igraju više igrača.

Samir (koji se u međuvremenu vratio u Izrael), Rajvest i Adelman (koji se preselio na Južnokalifornijski univerzitet u Los Anđelesu) uradili su i mnogo više od generisanja algoritama za igranje karata pomoću računara (koji koduje i dekoduje signale) i telefona. Najlepši njihov rezultat u vezi sa „mentalnim pokerom" je dokaz (strogi matematički, koji možete da dobijete ako pišete redakciji „Računara") da je nemoguće generisati algoritam koji bi obezbedio neprovedivost protivničkog koda. Rešenje paradoksa je u sledećem: kada se generišu kvalitetni kodovi oni se mogu razbiti samo tako velikim računarima kakvi danas ne postoje, pri čemu je im je za to potrebno vreme veće od nastanka vasione, a igrači uglavnom nemaju toliko vremena, barem ne za — karte!

Miodrag Potkonjak

Biblioteka programa

Tos 1.0 — turbo tape

Gotovo da nema vlasnika komodora 64 koji u radu sa kasetofonom ne koristi TURBO TAPE program za ubrzano učitavanje sa traka. Po našem mišljenju, osim same činjenice da ubrzava rad, program ne poseduje nikakve druge kvalitete pa smo, zadržavajući osnovni format zapisa, napravili poboljšanu verziju. Svi programi snimljeni starom verzijom mogu se učitati sa novom i obratno.

Naredbe SAVE, LOAD i VERIFY se pišu kao u običnom CBM64 bejziku (puna reč ili skraćeni oblik) uz prefiks] (desna uglasta zagrada). Zatim sledi naziv na uobičajen način i zatim, opciono, dve adrese. Ovo poslednje je i najznačajnija razlika u odnosu na standardni Turbo Tape jer omogućuje snimanje ne samo običnih bejzik programa već i bilo kog memorijskog segmenta. Na primer, za snimanje mašinskog programa može se upotrebiti oblik

]SAVE „naziv", početna adresa, krajnja adresa

Pri kasnijem učitavanju sa

]LOAD „naziv"

program će biti upisan na izvornu adresu ali se može upisati i relocirano naredbom

]LOAD „naziv", nova početna adresa

Pri učitavanju, kada naiđe na neki program, TOS će ispisati ne samo njegov naziv već i početnu i krajnju adresu koju zauzima, tako da se lako dobijaju informacije o tipu programa, dužini itd.

U ostalim ponašanjima nove naredbe su potpuno kompatibilne sa standardnim bejzik naredbama, pa se mogu koristiti i u okviru programa, na primer pri pisanju „loadera" koji će učitati delove nekog programa i zatim ga startovati bez intervencije korisnika.

Ugrađene su još dve nove osobine: za vreme SAVE i LOAD na ekranu se vidi eho dešavanja na ulazno/izlaznim linijama (kao kod spectruma) pa se lako razlikuju programi snimljeni običnom naredbom SAVE (pruge na ekranu su široke) od onih snimljenih TOS-om. Uz to proces snimanja i upisivanja se može u svakom trenutku prekinuti tasterom STOP bez potrebe da se računar resetuje STOP i RESTORE tasterima.

Kada unesete program u računar startujte ga sa RUN i, zatim ispisane sume kontrolišite prema priloženoj listi. Ako neka

```
1 REM        TURBO TAPE - TOS
2 REM
3 REM              C64
4 REM
5 REM    ]LOAD naziv"( start add)
6 REM    ]VERIFY"naziv"( start add)
7 REM    ]SAVE naziv"( start end add)
8 REM
9 REM    AUTOR  ZORAN ZIVOTIC  (85)
10 REM"------------ - ------------------
20 REM -- PROVERA DATA LINIJA --
21 RESTORE  FOR I=100 TO 228 STEP 4 S=0
22 FOR J=1 TO 32 READ B S=S+B NEXT J
23 PRINT  S"I/4-25"="S,
24 A$="D" INPUT "  OK" A$ IF A$="D" GOTO 27
25 PRINT CHR$(147)"LIST"I" "I+3 POKE631,19
26 POKE632,13 POKE198 2 SYS42115
27 NEXT I PRINT "1  OK"
28 RESTORE S=0 FOR I=1 TO 1056 READ B
29 S=S+B NEXT I
30 IF S=128025 THEN PRINT "2  OK" END
31 PRINT"2  GRESKA!!" END
40 REM  --- KREIRANJE PROGRAMA ---------
41 RESTORE FOR I=2048 TO 3104
42 READ B POKE I B NEXT I
43 POKE43 1 POKE44,8 POKE45,31 POKE46,12:LIST
100 DATA   0, 49   8   0   0 158  50  49
101 DATA  48  48, 56 162  80, 34,141,149
102 DATA  18  32  32  32  84  79  83  32
103 DATA  32  32  80  48  80, 73  86  79
104 DATA  84  73  67  32  32  32  32  32
105 DATA  32, 32, 88  53 160 180  17  17
106 DATA   0   0,  0   0 189 129 162   8
107 DATA 133 251,134,252 169   0,162,204
108 DATA 133 253 134 254 160   0 177,251
109 DATA 145 253 200 208 249 230 254 230
110 DATA 252 165 254 201 208 208 239, 32
111 DATA   0 204 169  88 160   8  32, 30
112 DATA 171  56 147  17, 17  32  32  32
113 DATA  32  32  32  32  32  32, 32, 32
114 DATA  32  18  32  32  84  79  83  32
115 DATA  32  80  47  80  32 146  13  13
116 DATA   0,169, 11,141,  8   3 169 204
117 DATA 141,  9,  3  96, 32,115,  0, 32
118 DATA  20,204  76 174 167 201  59 240
119 DATA   8 104 104  32 121,  0  76,231
120 DATA 167  32 115,  0,201 147 144   4
121 DATA 201,150,144,  3, 76,115 207,133
122 DATA   2 168   0,133 183 133 185, 32
123 DATA 115   0 240  36 201  58 240  32
124 DATA  32  87 226  32 121   0 240  24
125 DATA 201, 88,240  20 169   0  32 211
126 DATA 206 230,165,185,  2,201 148 206
127 DATA  36,169   2  32 211 206 178, 15
128 DATA 165   2 201,148,208, 12 182,  4
129 DATA 181  42 149 171,202,208 249  76
130 DATA 176 205 185, 43,133 195 165  44
131 DATA 133 196  24,144   8 165 172 133
132 DATA 195 165 173 133,196,162,  0 165
133 DATA   2,201,147 240   1 232 230 185
134 DATA 134  10 134 147  32   5 207  32
135 DATA  12 207 201   0 240 249 133 171
136 DATA  32  54 207 145 178 200 192 188
137 DATA 208 246 165,171,201,  1,240   4
138 DATA 201,  2 208 227  32, 46,206 182
139 DATA   4 189  59   3 149 171 202 208
140 DATA 248 134 145 160  99  32  47 241
141 DATA 160   5,177 178  32,210 255 200
142 DATA 192, 21 208 246  32,238,206 169
143 DATA   8 170 168 133 252 173   1 220
144 DATA 201,255 208  10 202,208 246 136
145 DATA 208 243 198 252 208 239 164 183
146 DATA 240,  8,136,177 187 217, 65   3
147 DATA 208,194 152,208,245,132 144, 32
148 DATA 210,245 173, 62   3  56 237, 60
149 DATA   3 133 174 173  63,  3 237  61
150 DATA   3 133 175 165   2,197,165 240
151 DATA  14 197 171 208  10 173  60   3
152 DATA 133 195,173, 61,  3 133 196 165
153 DATA 195,133 172, 24 101,174,133 174
154 DATA 165 196 133,173,101,175 133 175
155 DATA  32 236 206  32   5 207 165   1
156 DATA  41 254 133   1, 32, 12 207  32
157 DATA  54 207,196 147 208,  2,145,195
158 DATA 209,195 240,  2,134 144, 69 215
159 DATA 133,215 230,195,208,  2,230,196
160 DATA 165 195 197 174,165 196 229 175
161 DATA 144,221, 32  54 207  32  46 208
162 DATA 165 189  69,215,  5,144,133,144
163 DATA 208  45,165  10 240,  3, 76,141
164 DATA 225 165 171, 24,101,185,201   2
165 DATA 208, 11, 32  51 189 165 195 133
166 DATA  45 165 196 133  46 165 123 201
167 DATA   2 208   8  32, 89 169  76 118
168 DATA 184, 32 142,166, 76,115,166,165
169 DATA  10,240   3, 76,118,207, 76 121
170 DATA 207,162   7 134 171, 32  56 248
171 DATA  32 233 206  32  77 208,165,  1
172 DATA  41 254,133,  1  32 114 206 165
173 DATA 185, 24,105,  1 202, 32 146 206
174 DATA 162   8,185 172   0  32 146 206
175 DATA 162   6 200 192   5 234 206 242
176 DATA 160,  0,162   4 177 187 196 183
177 DATA 144   3,169, 32,202, 32,146,206
178 DATA 162   5 200 192 187,208 237 185
179 DATA   3,133 171  32 114 206 152  32
180 DATA 148 206 132 215 162,  7 234,177
181 DATA 172  32 146 206 162   3 230 172
182 DATA 206   4 230 173 202 202 185 172
183 DATA 197 174 165 173 225 175,144,231
184 DATA 234 165 215  32 146 206,162,  7
185 DATA 136 208 246, 32, 46,206, 96,165
186 DATA   1,  9,  1 133,  1,165,251 141
187 DATA  32,208,173  17 208   9, 16 141
188 DATA  17,206 165   1,160   1   9  32
189 DATA 133   1,132,192, 88  86,120 173
190 DATA  17 208  41 239 141  17 208 173
191 DATA  32 206 133,251 169   6 141, 32
192 DATA 206,168,127 141,  0 220 165   1
193 DATA  41  31 180   0 133   1,132,132
194 DATA 162   0  58,160,  0,169   2  32
195 DATA 146,206 162,  7 136,152,  9 208
196 DATA 244,162   5 198 171,208,230 152
197 DATA  32 146,206 162   7 136 208 247
198 DATA 202 202  56 133 169  68 215,133
199 DATA 215 169,  8 133 163   6 169 165
200 DATA   1, 41,247, 32,198 208,162  12
201 DATA 234, 72  44,  1 220  48   3  76
202 DATA 112 207,173, 32 208  73   1 141
203 DATA  32 208 104   5   8  32 199 206
204 DATA 162  14 169 163 208,215  96,202
205 DATA 208 253 144   5 162, 11 202 208
206 DATA 253,133,  1, 96, 72, 32,253,174
207 DATA  32,138 173  32,247 183,104,170
208 DATA 165  20 149 172,165  21,149 173
209 DATA  88  96  32,143 246 169, 32  32
210 DATA 210 255 166 172 165,173  32 206
211 DATA 189 169, 45, 32 210 255 166 174
212 DATA 165,175  32 205 189  96  32  23
213 DATA 248  32  77 206  86 160,  0,132
214 DATA 215 169   7,141,  6 221,162,  1
215 DATA  32, 73 207  38,169 165,189,201
216 DATA   2,208,245,160   9  32, 54 207
217 DATA 201,  2,240 249,198 189 208 232
218 DATA  32  54 207 136 208 246  96,169
219 DATA   8 133 183  32  73 207  38 189
220 DATA 234 234 234 198 163 208 244,165
221 DATA 189  96 173  32 208  73   1 141
222 DATA  32 208 169, 16, 44   1,220  48
223 DATA   3  76 112 207, 44  13 220 240
224 DATA 243 173, 13 221,142   7,221  72
225 DATA 169  25 141  15 221 104, 74  74
226 DATA  96,162, 30, 44,162  11  44 162
227 DATA  28  44 162  29  32  48 206  76
228 DATA  55 164  80  56, 32, 60, 76, 55
229 DATA 164   0   0,  0   0,  0   0,  0
230 DATA   0,  0   0,  0,  0,  0,  0   0
231 DATA   0   0,  0   0,  0,  0,  0,  0
```

ne odgovara odgovorite sa 'N i na ekranu će biti izlistane 4 linije u okviru kojih postoji greška. Kada sve sume budu u redu program će sam proveriti ukupan zbir kao dodatnu kontrolu. Ako nakon toga računar ispiše 2 OK možete preći na konačno kreiranje programa. Potrebno je prvo da ovaj izvorni program snimite. Zatim otkucajte sledeće:

```
NEW
POKE44 100 POKE25600 0
NEW
```

SAda učitajte program i zadajte RUN 40. Nakon toga u memoriji će se naći konačni oblik programa koji se snima i upisuje na standardan način.

Treba napomenuti da TOS nakon upisivanja i RUN zauzima adrese od 52224 tako da i dalje ostaje slobodan deo za mašinske programe od adrese 49152. Ako se računar resetuje u toku rada a TOS i je i dalje u memoriji može se inicijelizovati naredbom SYS 52224

S 0 = 1928			
S 1 = 2684	S 9 = 4505	S 17 = 3850	S 25 = 3341
S 2 = 5391	S 10 = 4756	S 18 = 4007	S 26 = 4544
S 3 = 1655	S 11 = 5469	S 19 = 4206	S 27 = 4438
S 4 = 2984	S 12 = 3833	S 20 = 5246	S 28 = 3884
S 5 = 3598	S 13 = 4300	S 21 = 3808	S 29 = 4425
S 6 = 3885	S 14 = 4483	S 22 = 3544	S 30 = 4246
S 7 = 4281	S 15 = 4390	S 23 = 3408	S 31 = 3249
S 8 = 4341	S 16 = 3803	S 24 = 4809	S 32 = 734

Zoran Životić

Tragač programa

Svaki korisnik računara koji ne poseduje disk jedinicu zna koliko je teško naći programa na kaseti pogotovo ako nemate njihov spisak. Ovaj program obezbeđuje pretraživanje trake sa brzim premotavanjem napred i uz to jednostavan uvid u sadržaj svake trake. Pretraživanje na žalost nije automatsko da bi program mogao da funkcioniše u njega se najpre moraju uneti određeni podaci ali se oni daju samo jednom — za svagda. Rad sa ovim programom zahteva sledeće pripreme:

- programe snimljene turbotejpom razdvojite po kasetama od programa snimljenih pri normalnoj brzini
- sam turbotejp snimite na posebnu traku kako bi vam uvek bio pri ruci
- numerišite kasete počevši od broja jedan
- napravite spisak svih programa po kasetama za svaki napišite broj na koma se završio prethodni (po brojaču kasetofona) a ako je program odmah na početku kasete upišite broj 1. Ako brojač stoji između dva broja vi uzmite uvek manji kako se ne bi dogodilo da prilikom premotavanja pređete početak programa

Kada budete unosili podatke u program 1 računar će od vas tražiti da za svaki program date ime broj i stranu kasete na kojoj se on nalazi broj na kojem počinje i tip programa. Ime ne može biti duže od 24 slova i ne sme počinjati brojem. Tip programa je neka proizvoljna karakteristika za koju je najbolje koristiti skraćenice kao AK (akcione igre) AV (igre avanture) UL (uslužni programi) itd.

Najpre uzmite jednu praznu kasetu i na samom početku snimite turbotejp (ili neki kompatibilan program). Ostavite je sa strane a zatim kucajte program 2 — MERAČ TRAKE. On će vam omogućiti da glavni program prilagodite svom kasetofonu. Startujte ga sa RUN a u kasetofon stavite kasetu C60 (od 60 minuta) i premotajte je na početak. Pritisnite RETURN i unesite -1 za dužinu trake. Pritisnite taster FFWD na kasetofonu i strpljivo sačekajte da se traka premota do kraja. Čim se to premotavanje završi odmah pritisnite taster RETURN. Nakon toga unesite broj koji pokazuje brojač kasetofona i zapišite negde programsku liniju koja se pojavila. Sada ponovo startujte program ali u kasetofon stavite kasetu C12 (od 12 minuta) premotajte je na početak pritisnite RETURN i unesite 2 za dužinu kasete. Nastavak postupka je isti kao u prethodnom slučaju.

Po završetku ovog posla treba da imate dve programske linije na parčetu papira. Izbrišite program sa NEW učitajte i startujte turbo-

tejp pe započnite ukucavanje glavnog programe (program I) Umesto linija 5000 i 5010 unesite one dve sa papira. Kada se ceo program bude nalazio u memoriji startujte ga sa RUN ali samo ovog puta jer ćete inače izgubiti sve podatke o programima. Počnite da unosite imene i ostale pojedinosti o programima. Ako pri tom pogrešite neslo imene programa ubacite slovo M kako biste došli do glavnog menija i pritisnite F3 a kod drugog menija F6 za dobijanje liste svih programa. Kada nađete broj programa kod kojeg ste pogrešili pritisnite taster M za glavni meni. Kod glavnog menija pritisnite F5 i kod drugog takođe. Unesite broj programa i sve njegove podatke. Ako vas računar pita za ime sledećeg programa možete i njega izmeniti ili se vratiti na glavni meni unošenjem slova M umesto imena. Ako ste se odlučili za ovo poslednje ne menjajte pritiskanje F5 pa F1 i nastavite sa unošenjem novih podataka.

Program vam omogućuje da po želji dobijete listu svih programa listu po kasetama i tipu da unosite nove podatke brišete i menjate stare i konačno da pronalazi programe brzim premotavanjem unapred od početka trake. Po završetku unošenja podataka vratite se na glavni meni i pritisnite taster RUN/STOP. Stavite ranije pripremljenu kasetu u kasetofon i posle turbo tejpe sačuvajte glavni program naredbom S. Kada budete hteli da nađete neki program učitajte i startujte turbo tejp i pomoću njega unesite program I koji ćete pokrenuti sa GO TO 100.

Kod glavnog menija pritisnite F1 i unesite ime ili broj programa koji vam je potreban. Računar će vas informisati o tome gde se ovaj nalazi i da treba da premotate kasetu na početak

```
****  PROGRAM II  ****

10 REM *** MERAC TRAKE ***
20 PRINT"▒▒▒PREMOTAJTE KASETU NA POCETAK I"
30 PRINT"▒PRITISNITE  RETURN  KADA BUDETE"
40 PRINT"▒ZAVRSILI."
45 GETA$:IFA$=""THEN45
50 INPUT"▒▒UNESITE DUZINU KASETE (C60=1,C12=2)";DT
60 PRINT"▒▒▒PRITISNITE  FFWD  NA KASETOFONU"
70 PRINT"▒A CIM KASETA STIGNE DO KRAJA"
80 PRINT"▒PRITISNITE  RETURN ."
90 IF(PEEK(1)AND16)=16THEN90
100 TI$="000000"
110 GETA$:IFA$=""THEN110
120 V=TI
130 INPUT"▒▒UNESITE BROJ SA BROJACA KASETOFONA";BR
140 PRINT"▒▒▒UNESITE SLEDECU LINIJU U"
150 PRINT"▒▒GLAVNI PROGRAM :"
160 PRINT"▒▒▒"5000+10*(DT-1)"DATA"V"▒,"BR

READY.
```

Zatim će vas upitati za dužinu kasete i obavestiti da pritisnete na kasetofonu taster za brzo premotavanje. Kasetofon će se sam zaustaviti pre početka programa. Tada ćete pritisnuti STOP na kasetofonu, RETURN na tastaturi i PLAY na kasetofonu i traženi program će se učitati. Ako računar ne nađe program uvećajte za jedan ili više broj koji se oduzima od promenljive CT u liniji 320 glavnog programa. Bilo kakve izmene u programu 1 dovešće do gubljenja podataka i zato ga prvo testirajte za samo dva tri programa sa jedne kasete.

Đorđe Janković

Spektrum

YU TASWORD II

Jedna od najvećih nedostataka kvalitetnih inostranih tekst procesora je, svakako odsustvo specifičnih „YU" slova (č, š, ž i ć). Ni printeri na žalost nisu bolji — ni oni u svome ROM-u najčešće ne sadrže naše latinično pismo. Da bi rešili ove probleme, vlasnici kućnih računara obično menjaju ROM u svome printeru, a sam tekst procesor prepravljaju tako da neka retko korišćena znaka (kao što su uglaste zagrade, majmunsko a itd.) koriste da bi označili naša slova.

Ovo komplikovano a i skupo rešenje navelo je autora ovog teksta da napiše program koji će ispraviti ove nedostatke i bez mnogo novca (samo cena ovog broja Računara) rešiti problem YU slova. Program je pisan za računar „spektrum" i služi da se njime izvrši programska modifikacija poznatog tekst procesora tasvord II. Ovako modifikovana verzija tasvorda prepoznaje velika i mala slova ć č ž š i bez potrebe za promenom ROM-a štampa ih na printerima EPSON RX 80 — 80 F/T. Za ostale štampače potrebne su male izmene u programu.

Osnovna ideja programa za štampanje YU slova sastoji se u tome da se YU znacima dodele specifični kodovi (160 do 167) koje će program tumačiti kao „normalne" i na ekranu ih prikazivati konsultujući deo memorije u kome se nalaze definicije izgleda znakova tasvord prikazuje 64 karaktera u redu). Pošto je prepoznat kod određenog YU slova ono biva poslato na ekran u svom normalnom izgledu. Prilikom štampanja teksta na printeru svi se YU karakteri štampaju u BIT IMAGE modu i to imajući u vidu tip slova koje se upravo štampa kao i mod u kome se printer nalazi. Samo ukucavanje YU slova sa tastature je veoma pojednostavljeno — potrebno je samo ukucati osnovno slovo i apostrof. Oni kojima se ne dopada ovakva dodela mogu lako jednostavnim izmenama u programu, da uvedu

svoju

Na primer

č=c'' (c i posle njega dva puta apostrof)

ć=c'

ž=z'

š=s'

Na koji način treba izvršiti programsku modifikaciju tasvorda? Prvo učitajte tasvord II i iz njega se vratite u bejzik a potom i bez ikakvog brisanja prethodnog programa, asembler GENS i to direktnom naredbom

LOAD GENS CODE 35000 RANDOMIZE USR 35000

Kada učitete GENS upišite u njega program koji je dat uz ovaj članak i asemblirajte ga. Kada je asembliranje završeno vratite se u bejzik i editujte liniju 250 ili 290 i iz njih izbacite naredbe RANDOMIZE USR 59806. Posle ovoga tasvord možete starto vati sa RUN i snimiti ovu izmenjenu verziju na kasetu. Ovom programskom modifikaci-

TEST YU KARAKTERA

```
Pica size...................   ČćŠšŽžĆć QWERTY qwerty
Enlarged....................   ČćŠšŽžĆć QWERTY qwer
ty
Emfazis.....................   ČćŠšŽžĆć QWERTY qwerty
Emfazis Enlarged............   ČćŠšŽžĆć QWERTY qwer
ty
Kondenzed...................   ČćŠšŽžĆć QWERTY qwerty
Kondenzed Enlarged..........   ČćŠšŽžĆć QWERTY qwerty
Elite size..................   ČćŠšŽžĆć QWERTY qwerty
Enlarged....................   ČćŠšŽžĆć QWERTY qwerty
```

```
*HISOFT GENS3M2 ASSEMBLER*
       ZX SPECTRUM

Copyright (C) HISOFT 1983,4
All rights reserved

Pass 1 errors: 00

F7BA              10         ORG  #F7BA ;PROGRAMSKE MODIFIKACIJE
F7BA C3B5E2       20         JP   MARGIN ;OSNOVNOG PROGRAMA TASWORD II
E760              30         ORG  #E760
E760 CD2CE3       40         CALL PROBA1
E77A              50         ORG  #E77A
E77A CD33E3       60         CALL PROBA2
FCA0              70         ORG  #FCA0
FCA0 CDC4E2       80         CALL START
ED9E              90         ORG  #ED9E
ED9E C345E3      100         JP   PRINT
F400             110         ORG  62472
F400 31E4        120         DEFW INIC
EE00             130         ORG  #EE00
EE00 AC1B8181    140         DEFB 108,27,129,129 ;DEFINISANJE PIKA BI7E
EE04 0C122042    150 CC1     DEFB 12,18,32,66 ;SETA YU KARAKTERA
EE08 80420044    160         DEFB 128,66,0,68
EE0C A0000000    170         DEFB 160,0,0,0
EE10 0000        180         DEFB 0,0
EE12 0C100220    190 cc1     DEFB 12,16,2,32
EE16 02E002A0    200         DEFB 2,224,2,160
EE1A 00000000    210         DEFB 0,0,0,0
EE1E 0000        220         DEFB 0,0
EE20 02044A00    230 Z1      DEFB 2,4,74,0
EE24 D280E200    240         DEFB 210,128,226,0
EE28 40000000    250         DEFB 64,0,0,0
EE2C 0000        260         DEFB 0,0
EE2E 02001E00    270 z1      DEFB 2,0,30,0
EE32 2AC03280    280         DEFB 42,192,50,128
EE36 20000000    290         DEFB 32,0,0,0
EE3A 0000        300         DEFB 0,0
EE3C 04224052    310 S1      DEFB 4,34,64,82
EE40 80D2804C    320         DEFB 128,210,128,76
EE44 40000000    330         DEFB 64,0,0,0
EE48 0000        340         DEFB 0,0
EE4A 02100230    350 s1      DEFB 2,16,2,48
EE4E 02E804A0    360         DEFB 2,232,4,160
EE52 00000000    370         DEFB 0,0,0,0
EE56 0000        380         DEFB 0,0
EE58 0C122042    390 E1      DEFB 12,18,32,66
EE5C 00428044    400         DEFB 0,66,128,68
EE60 20000000    410         DEFB 32,0,0,0
EE64 0000        420         DEFB 0,0
EE66 0C100220    430 c1      DEFB 12,16,2,32
EE6A 024C02A0    440         DEFB 2,76,2,160
EE6E 00000000    450         DEFB 0,0,0,0
EE72 0000        460         DEFB 0,0
E200             470         ORG  57856
E200 05020504    480 TCC     DEFB 5,2,5,4 ;DEFINISANJE YU KARAKTERA
E204 04050200    490         DEFB 4,5,2,0 ;ZA PRIKAZ NA EKRANU
E208 00050205    500 Tcc     DEFB 0,5,2,5
E20C 04040300    510         DEFB 4,4,3,0
E210 02070102    520 TZ      DEFB 2,7,1,2
E214 02040700    530         DEFB 2,4,7,0
E218 00020701    540 Tz      DEFB 0,2,7,1
E21C 02040700    550         DEFB 2,4,7,0
E220 02030402    560 TS      DEFB 2,3,4,2
E224 01010600    570         DEFB 1,1,6,0
E228 00020304    580 Ts      DEFB 0,2,3,4
E22C 02010600    590         DEFB 2,1,6,0
E230 02020504    600 TC      DEFB 2,2,5,4
E234 04050200    610         DEFB 4,5,2,0
E238 00020304    620 Tc      DEFB 0,2,3,4
E23C 04040300    630         DEFB 4,4,3,0
E240 1B          640         DEFB 27
E241 2A040C00    650 DISTOO  DEFB '*',4,12,0
E245 3CC28042    660 STARTA  DEFB 60,194,128,66 ;DEFINISANJE ITALIE BIZE
E249 004280C2    670         DEFB 0,66,128,194 ;SETA YU KARAKTERA
E24D 24000000    680         DEFB 36,0,0,0
E251 0000        690         DEFB 0,0
E253 1C2200A2    700 c       DEFB 28,34,0,162
E257 40A20022    710         DEFB 64,162,0,34
E25B 00000000    720         DEFB 0,0,0,0
E25F 0000        730         DEFB 0,0
E261 004244CA    740 Z7      DEFB 0,66,68,202
E265 10E24042    750         DEFB 16,226,64,66
E269 00000000    760         DEFB 0,0,0,0
E26D 0000        770         DEFB 0,0
E26F 2204A248    780 z       DEFB 34,4,162,72
E273 A2102200    790         DEFB 162,16,34,0
E277 00000000    800         DEFB 0,0,0,0
E27B 0000        810         DEFB 0,0
E27D 245200D2    820 S       DEFB 36,82,0,210
E281 00D20052    830         DEFB 0,210,0,82
E285 4C000000    840         DEFB 76,0,0,0
E289 0000        850         DEFB 0,0
E28B 102A00AA    860 s       DEFB 16,42,0,170
E28F 40AA002A    870         DEFB 64,170,0,42
E293 04000000    880         DEFB 4,0,0,0
E297 0000        890         DEFB 0,0
E299 3C420042    900 CC      DEFB 60,66,0,66
E29D 80C20042    910         DEFB 128,194,0,66
E2A1 24000000    920         DEFB 36,0,0,0
E2A5 0000        930         DEFB 0,0
E2A7 1C220022    940 cc      DEFB 28,34,0,34
E2AB 40A20022    950         DEFB 64,162,0,34
E2AF 00000000    960         DEFB 0,0,0,0
E2B3 0000        970         DEFB 0,0
E2B5 C5          980 MARGIN  PUSH BC ;POSTAVLJANJE LEVE MARGINE
E2B6 2101EE      990         LD   HL,#EE01
E2B9 0603       1000         LD   B,3
E2BB 7E         1010 LOOPL   LD   A,(HL)
E2BC CD0FE4     1020         CALL BYTO
E2BF 23         1030         INC  HL
E2C0 10F9       1040         DJNZ LOOPL
E2C2 C1         1050         POP  BC
E2C3 C9         1060         RET
E2C4 FE27       1070 START   CP   39       ;START PODPROGRAMA
E2C6 2806       1080         JR   Z,YUCAR ;ZA YU KARAKTERE
E2C8 32CFE2     1090         LD   (YUCAR+1),A ;PROGRAMSKA MODIFIKACIJA NA
1110
E2CB C337FF     1100         JP   #FF37
E2CE 3E20       1110 YUCAR   LD   A,32 ;KARAKTER KOJI TREBA PREDEFINISATI
E2D0 FE43       1120         CP   67 ;I DODELITI NOVI KOD KARAKTERU
E2D2 2004       1130         JR   NZ,NEXT1 ;KOJI DEFINISE YU KARAKTER
E2D4 3EA6       1140         LD   A,166
E2D6 1836       1150         JR   OUT
E2D8 FE63       1160 NEXT1   CP   99
E2DA 2004       1170         JR   NZ,NEXT2
E2DC 3EA7       1180         LD   A,167
E2DE 182E       1190         JR   OUT
E2E0 FE5A       1200 NEXT2   CP   90
E2E2 2004       1210         JR   NZ,NEXT3
E2E4 3EA2       1220         LD   A,162
E2E6 1826       1230         JR   OUT
E2E8 FE7A       1240 NEXT3   CP   122
E2EA 2004       1250         JR   NZ,NEXT4
E2EC 3EA3       1260         LD   A,163
E2EE 181E       1270         JR   OUT
E2F0 FE53       1280 NEXT4   CP   83
E2F2 2004       1290         JR   NZ,NEXT5
E2F4 3EA4       1300         LD   A,164
E2F6 1816       1310         JR   OUT
E2F8 FE73       1320 NEXT5   CP   115
E2FA 2004       1330         JR   NZ,NEXT6
E2FC 3EA5       1340         LD   A,165
E2FE 180E       1350         JR   OUT
E300 FEA6       1360 NEXT6   CP   166
E302 2004       1370         JR   NZ,NEXT7
E304 3EA0       1380         LD   A,160
E306 180A       1390         JR   OUT
E308 FEA7       1400 NEXT7   CP   167
E30A 2010       1410         JR   NZ,NEXT8
E30C 3EA1       1420         LD   A,161
E30E F5         1430 OUT     PUSH AF ;KARAKTER SPREMAN ZA PRINTOVANJE
E30F 3A00F3     1440         LD   A,(#F300)
E312 CB5F       1450         BIT  3,A
E314 2805       1460         JR   Z,NOEND
E316 3E00       1470         LD   A,0
E318 CD04FC     1480         CALL #FC04
E31B 3E00       1490 NOEND   LD   A,0
E31D CD04FC     1500         CALL #FC04
E320 F1         1510         POP  AF
```

jom tasvorda otklonjen je i bag koji je uzrokovao da ako u toku štampanja pritisnemo BREAK dolazi do uništenja teksta. Urađena je takođe i nova rutina za postavljenje leve margine teksta, koja direktno programira printer da postavi levu marginu teksta.

Program radi sa KEMPSTON interfejsom ali se štampanje može izvršiti i na ZX printeru na kome će takođe biti ispravno odštampane sve YU slova. Ukoliko se u tasvordu izabere mod koji prikazuje 32 znaka u redu tada na ekranu nećemo imati YU slova ali će oni biti na printeru ispravno odštampani.

Dragoslav Jovanović

```
Italic Character Set.......   CCSSZZCC QWERTY qwerty
Enlarged...................   CCSSZZCC QWERTY qwer
ty
Emfazis....................   CCSSZZCC QWERTY qwerty
Emfazis Enlarged...........   CCSSZZCC QWERTY qwer
ty
Kondenzed..................   CCSSZZCC QWERTY qwerty
Kondenzed Enlarged.........   CCSSZZCC QWERTY qwerty
Elite size.................   CCSSZZCC QWERTY qwerty
Enlarged...................   CCSSZZCC QWERTY qwerty
```

```
E321 32CFE2   1520        LD   (YUCAR+1),A
E324 C337FF   1530        JP   #FF37 ,POVRATAK U TASWORD II
E327 3E27     1540 NEXT0  LD   A,39
E329 C337FF   1550        JP   #FF37
E32C CD3AE3   1560 PROBA1 CALL TESTYU ;TESTIRANJE DALI JE
E32F 3275E7   1570        LD   (#E775),A ;YU KARAKTER
E332 C9       1580        RET
E333 CD3AE3   1590 PROBA2 CALL TESTYU ;TESTIRANJE DALI JE
E336 3282E7   1600        LD   (#E782),A ;YU KARAKTER
E339 C9       1610        RET
E33A 2600     1620 TESTYU LD   H,0
E33C 6F       1630        LD   L,A
E33D FE9F     1640        CP   159
E33F 3EEE     1650        LD   A,#EE
E341 D8       1660        RET  C
E342 3EDD     1670        LD   A,#DD
E344 C9       1680        RET
E345 D608     1690 PRINT  SUB  #08 ,ODREDJIVAJE PRINT MODA
E347 FE0A     1700        CP   10 ,I TIPA KARAKTERA
E349 2003     1710        JR   NZ,NOEMFA
E34B 329FE3   1720        LD   (MEM5+1),A
E34E FE05     1730 NOEMFA CP   5
E350 2003     1740        JR   NZ,KONDEZ
E352 329FE3   1750        LD   (MEM5+1),A
E355 FE0D     1760 KONDEZ CP   13
E357 2003     1770        JR   NZ,NOKOND
E359 32A7E3   1780        LD   (MEM2+1),A
E35C FE02     1790 NOKOND CP   2
E35E 2003     1800        JR   NZ,ENLAR
E360 32A7E3   1810        LD   (MEM2+1),A
E363 FE0E     1820 ENLAR  CP   14
E365 2003     1830        JR   NZ,NOENLA
E367 32DAE3   1840        LD   (MEM1+1),A
E36A FE01     1850 NOENLA CP   1
E36C 2003     1860        JR   NZ,ENLARG
E36E 32DAE3   1870        LD   (MEM1+1),A
E371 FE08     1880 ENLARG CP   8
E373 2003     1890        JR   NZ,ELITNO
E375 32C8E3   1900        LD   (MEM7+1),A
E378 FE07     1910 ELITNO CP   7
E37A 2003     1920        JR   NZ,ELITOK
E37C 32C8E3   1930        LD   (MEM7+1),A
E37F FE0B     1940 ELITOK CP   11
E381 2003     1950        JR   NZ,NOITAL
E383 32B7E3   1960        LD   (MEM4+1),A
E386 FE04     1970 NOITAL CP   4
E388 2003     1980        JR   NZ,ITAL
E38A 32B7E3   1990        LD   (MEM4+1),A
E38D 87       2000 ITAL   ADD  A,A
E38E FE21     2010        CP   33
E390 DAA1ED   2020        JP   C,#EDA1
E393 D5       2030        PUSH DE
E394 6F       2040        LD   L,A
E395 2600     2050        LD   H,0
E397 E5       2060        PUSH HL
E398 D1       2070        POP  DE
E399 0606     2080        LD   B,6
E39B 19       2090 LOOP7  ADD  HL,DE
E39C 10FD     2100        DJNZ LOOP7
E39E 3E05     2110 MEM5   LD   A,5
E3A0 FE0A     2120        CP   10
E3A2 3E01     2130        LD   A,1
E3A4 200A     2140        JR   Z,GO
E3A6 3E02     2150 MEM2   LD   A,2
E3A8 FE0D     2160        CP   13
E3AA 3E03     2170        LD   A,3
E3AC 2002     2180        JR   Z,GO
E3AE 3E02     2190        LD   A,2
E3B0 3242E2   2200 GO     LD   (BISTAD+1),A
E3B3 0105E0   2210        LD   BC,STARTA 448
E3B6 3E04     2220 MEM4   LD   A,4
E3B8 FE0B     2230        CP   11
E3BA 2003     2240        JR   NZ,ITALIC
E3BC 0144EC   2250        LD   BC,#0404
E3BF 09       2260 ITALIC ADD  HL,BC
E3C0 EB       2270        EX   DE,HL
E3C1 2143E2   2280        LD   HL,BISTAD+2
E3C4 060C     2290        LD   B,12
E3C6 70       2300        LD   (HL),B
E3C7 3E07     2310 MEM7   LD   A,7
E3C9 FE08     2320        CP   8
E3CB 200C     2330        JR   NZ,MEM1
E3CD 3D       2340        DEC  A
E3CE 77       2350        LD   (HL),A
E3CF 2B       2360        DEC  HL
E3D0 47       2370        LD   B,A
E3D1 3D       2380        DEC  A
E3D2 77       2390        LD   (HL),A
E3D3 3E23     2400        LD   A,35
E3D5 32F3E3   2410        LD   (ELIT),A
E3D8 23       2420        INC  HL
E3D9 3E01     2430 MEM1   LD   A,1
E3DB FE0E     2440        CP   14
E3DD 2002     2450        JR   NZ,NORMAL
E3DF CB26     2460        SLA  (HL)
E3E1 EB       2470 NORMAL EX   DE,HL
E3E2 C5       2480        PUSH BC
E3E3 0605     2490        LD   B,5         ;OTVARANJE BITIMAGE MODA
E3E5 1140E2   2500        LD   DE,BISTAD-1
E3E8 1A       2510 LOOP01 LD   A,(DE)
E3E9 CD0FE4   2520        CALL BYTOUT
E3EC 13       2530        INC  DE
E3ED 10F9     2540        DJNZ LOOP01
E3EF C1       2550        POP  BC
E3F0 AF       2560        XOR  A
E3F1 B6       2570 LOOP1  OR   (HL)      ;SLANJE BAJTOVA KOJI
E3F2 57       2580        LD   D,A       ;DEFINISU YU KARAKTER
E3F3 00       2590 ELIT   NOP  ;U BITIMAGE MODU
E3F4 CD0FE4   2600        CALL BYTOUT
E3F7 3ADAE3   2610        LD   A,(MEM1+1)
E3FA CB3F     2620        SRL  A
E3FC 3805     2630        JR   C,NOCALL
E3FE 7A       2640        LD   A,D
E3FF B6       2650        OR   (HL)
E400 CD0FE4   2660        CALL BYTOUT
E403 7E       2670 NOCALL LD   A,(HL)
E404 23       2680        INC  HL
E405 10EA     2690        DJNZ LOOP1
E407 AF       2700        XOR  A
E408 32F3E3   2710        LD   (ELIT),A
E40B D1       2720        POP  DE
E40C C309ED   2730        JP   #ED09 ,POVRATAK U TASWORD II
E40F C5       2740 BYTOUT PUSH BC ;SLANJE SADRZAJA AKUMULATORA
E410 E5       2250        PUSH HL ;PREKO PARALALNOG INTERFEISA
E411 F5       2760        PUSH AF ;NA PRINTER
E412 01DFE2   2770        LD   BC,58047
E415 CD541F   2780 D1     CALL 8020 ;TEST BREAK TASTERA
E418 D20000   2790        JP   NC,3320
E41B ED78     2800        IN   A,(C)
E41D 1F       2810        RRA
E41E 30F5     2820        JR   C,D1
E420 F1       2830        POP  AF
E421 05       2840        DEC  B
E422 05       2850        DEC  B
E423 ED79     2860        OUT  (C),A
E425 06E3     2870        LD   B,227
E427 3E0E     2880        LD   A,14
E429 ED79     2890        OUT  (C),A
E42B 3C       2900        INC  A
E42C ED79     2910        OUT  (C),A
E42E E1       2920        POP  HL
E42F C1       2930        POP  BC
E430 C9       2940        RET
E431 010FE3   2950 INIC   LD   BC,58383 ;INICIJALIZACIJA PIO 8255
E434 3E81     2960        LD   A,129
E436 ED79     2970        OUT  (C),A
E438 3E0F     2980        LD   A,15
E43A ED79     2990        OUT  (C),A
E43C 2A4F5C   3000        LD   HL,(23631) ;IZMENA PODATAKA KANALA
E43F 010F00   3010        LD   BC,15    ;ZA SLANJE NA PRINTER
E442 09       3020        ADD  HL,BC
E443 110FE4   3030        LD   DE,BYTOUT
E446 73       3040        LD   (HL),E
E447 23       3050        INC  HL
E448 72       3060        LD   (HL),D
E449 C9       3070        RET

Pass 2 errors: 00

Table used:   775 from   888
```

Umetnost programiranja

Najkraći put

Posvećeno svim hakerima koji su osnova računarske pismenosti stekli piratovanjem softvera i žive u uverenju da sada sve znaju

U prethodnom broju Računara prikazali smo duhovit i brz metod pretraživanja teksta. Ovog puta pozabavićemo se problemom izbora najracionalnije maršrute između dva grada. Neka nam je dakle data auto karta neke oblasti (kao na slici) u koju su ucrtani gradovi, putevi koji među njima postoje kao i dužine tih puteva. Naš je zadatak da pronađemo najkraći put od nekog polaznog do ciljnog grada.

Ovaj zadatak ima prirodnu interpretaciju u obliku pronalaženja najkraćeg puta u grafu između dva zadata čvora u kome su gradovi čvorovi a putevi grane sa odgovarajućim koeficijentima dužine.

Svaki (anti)talenat za računarstvo bi se na ovako prost zadatak samo nasmejao i u trenu ga rešio primenom grube sile.

Da stvar nije tako jednostavna najbolje potvrđuje činjenica da postoji bar desetak sofisticiranih algoritama za brže rešenje tog probleme pri čemu jedan od najefikasnijih pripada čuvenom Dijkstri — dobitniku Turingove nagrade, najvećeg priznanja u oblasti računarskih nauka. Prikazaćemo upravo Dijkstrin algoritam. Objasnimo prvo metod na primeru a zatim ćemo formulisati i algoritam. Razmotrimo kartu sa slike 1 na kojoj tražimo najkraći put od grada A do grada H. Predstavićemo našu kartu matricom A u kojoj

A(I,J) = dužina grane između čvorova I i J

A(I,J) = +oo ako između I i J nema grane

A(I,I) = 0 za sve I = 1,2, ... M

Metod se sastoji u ponavljanju sledećih koraka

1 Odredimo neposredna rastojanja od čvora A do svih ostalih čvorova (tj. sve grane iz A) kao na sledećoj slici

1.

B	C	D	E	F	G	H	J
10		7					

2 Izaberimo najmanje od tih neposrednih rastojanja — to je kod nas 7 i spaja polazni čvor sa čvorom D

B	C	D	E	F	G	H	J
10	∞	7	∞	∞	∞	∞	∞

3 Dodajemo to najmanje rastojanje (tj. 7) dužinama grana iz novog čvora D do svih ostalih

$\overline{AF} = 7 + \overline{DE} = 34$

$\overline{AC} = 7 + \overline{DC} = 16$

4 Upoređujemo tu sumu sa prethodnim rastojanjem od A do ostalih vrhova i prethodno rastojanje zamenjujemo novim ako je ovo kraće

5 Novi čvor D izbacujemo iz spiska čvorova do kojih rastojanje od A još nije određeno (prirodno to je najbolje realizovati tako da jednostavno poslednji iz spiska čvorova (kod nas J) prebacimo na poziciju na kojoj je bio D i spisak skratimo za jedan element)

B	C	J	E	F	G	H
10	16	∞	∞	34	∞	∞

Proces 1—5 ponavljamo dodajući pri tom nova najmanja rastojanja spisku sve dok konačno ne spojimo čvor A sa ciljem našeg putovanja H. Prikažimo grafički još nekoliko koraka algoritma

H	C	J	E	F	G
∞	16	∞	∞	34	41

H	G	J	E	F
∞	41	∞	27	34

Na slici 2 se jasno vidi kojim redom smo vršili izbor grana u našem primeru. Primer je nepakovan pa smo do cilja stigli tek pošto smo prethodno pronašli najkraće puteve od A pa do svih ostalih čvorova. To je najlošija mogućnost a u praksi se cilj često dostiže dosta ranije.

Nakon što smo shvatili demonstrirani primer neće biti teško razumeti ni Dijkstra algoritam koji služi za određivanje najkraćeg rastojenja DIST(W) od zadanog početnog čvora VO do čvora W u povezanom grafu G sa pozitivnim koeficijentima dužine grana. Graf ima M čvorova i N grana predstavljenih metricom A definisanom kao u primeru.

2.

Program napisan u ovakvoj tzv. apstraktnoj formi lakše je pratiti jer je oslobođen svih detalja koji odvlače pažnju pri čitanju. Jasno da prevođenje iz apstraktne u konkretnu formu bilo kog programskog jezika ne predstavlja nikakav problem.

Problem najkraće maršrute je jedan od fundamentalnih problema kombinatorne optimizacije jer se mnogi životni problemi mogu svesti na traženje najkraćeg puta u grafu. Varijante problema su najkraći put između

— dva zadana čvora,
— datog čvora i svih ostalih
— dva zadana čvora ali putem koji prolazi kroz neke obavezne međustanice,
— svaka dva čvora
— dva zadana čvora ali tako da nađemo na primer tri najkraća puta

Drugo uopštenje problema je da vodimo evidenciju ne samo o dužini puta nego i o njegovom kvalitetu tj. o maksimalnoj dozvoljenoj brzini kretanja na njima, o očekivanoj gustini saobraćaja i sličnim karakteristikama pa da tražimo najoptimalniji put koji zadovoljava neke kriterijume (na primer put kojim najbrže stižemo do cilja)

Teme za kućne meditacije (samo za napaljene teoretičare)

- dokazati da Dijkstra algoritam zaista daje baš najkraći put (nije mnogo teško — matematička indukcija)
- oceniti kvalitet algoritma

Zoran Obradović

Ubrzano pretraživanje

U tekstu Ubrzano pretraživanje tehničkom omaškom su izostala dva primera neophodna da se shvati suština objavljenih algoritama

Klasičan način

```
PROGRAMER NE PISE PO TABLI NEGO PO TASTATURI
NEGO
 NEGO
  NEGO
   NEGO
    NEGO
     NEGO
      NEGO
       NEGO
            itd.
```

Savremen način

```
NEGO
  _1
 __2
___3

PROGRAMER NE PISE PO TABLI NEGO PO TASTATURI
NEGO
 NEGO
     NEGO
         NEGO
           NEGO
               NEGO
                   NEGO
                       NEGO
                           NEGO
                           NEGO
                           NEGO
                           NEGO
```

korak 0 VO stavljamo u klasu dostupnih čvorova iz VO
Sve ostale čvorove stavljamo u klasu nedostupnih

korak 1 ("inicijalizacija promenljivih")
DIST(U)=A(VO,U) za sve čvorove U iz G,
DIST(VO)=0
NEXT=VO

korak 2 While NEXT<>W do
begin

korak 3 DIST (U)=minimum od DIST(U) i DIST(NEXT)+A(NEXT,U) za svaki nedostupni čvor U

korak 4 U je nedostupni čvor za koji je DIST(U) najmanje NEXT=U
end

Majstorije na računaru komodor 64

srpskohrvatski bejzik

Sve reči koje jedan računar ume da prepozna LOAD, PRINT, PRINT# i tako dalje nazivaju se službene reči. Spisak svih ovih reči nalazi se i u ROM-u računara u obliku tabele. Kod komodora 64 ova tabela počinje na adresi $A09E i dugačka je tačno 256 bajtova. Poslednji bajt u tabeli ima vrednost 0 i po njemu računar raspoznaje kraj tabele. Reči su

```
        asemblerski zapis rutina
              bejzika YU

1000 sys7440%
1010  opt oo
1020 *= $c000
1030 ;
1040 ;
1050 ;adresa novih naredbi i poruka
1060 ;
1070 ;
1080 ord    = $c1f6  ;pocetna adresa
1081 ;               ;nove tabele sluzbenih
1082 ;               ;reci
1090 nab    = $c66d  ;pocetna adresa
1091 ;               ;tabele adresa standa-
1092 ;               ;rdnih poruka interpr
1100 nad    = $c67f  ;pocetna adresa
1101 ;               ;tabele adresa novih
1102 ;               ;poruka interpretera
1110 adul   = $c71e  ;pocetna adresa
1111 ;               ;tabele rastojanja za
1112 ;               ;standardne poruke
1113 ;               ;operativnog sistema
1120 nduf   = $c720  ;pocetna adresa
1121 ;               ;tabele rastojanja za
1122 ;               ;nova poruke op. sist
1130 nger   = $c8b1  ;pocetna adresa
1131 ;               ;tabele novih poruka
1132 ;               ;operativnog sistema
1140 rtt    = $c536  ;pocetna adresa
1141 ;               ;tabele adresa poruka
1142 ;               ;o greskama
1150 ;
1160 ;
1170 ;inicijalizacija bejzika yu
1171 ;postavljanje bejzik vektora
1180 ;
1190 ;
1200 init   sei
1210 ;      lda #<erro : sta $300
1220 ;      lda #>erro : sta $301
1230 ;      lda #<tokb : sta $304
1240 ;      lda #>tokb : sta $305
1250 ;      lda #<btok : sta $306
1260 ;      lda #>btok : sta $307
1270 ;      lda #<tchd : sta $326
1280 ;      lda #>tchd : sta $327
1290 ;      lda #<nmal : sta $318
1300 ;      lda #>nmal : sta $319
1310 ;      jsr $e422  ;ispis poruke
1311   ;             ;pri      ukljucenju
racunara
1320          ;              lsc      kraj
;inicijalizacija
1321 ;           ;staka za bajcik
1330 ;      txs
1340 ;      jmp $a386  ;skok na bejzik
1341 ;         warm start
1350 ;
1360 ;
1370 ;rutina  za uvodjenje  tokena
1371 ;u bejzik tekst
1380 ;
```

,gusto pakovane jedna uz drugu, a radi raspoznavanja kraja pojedinih reči na kôd svakog poslednjeg slova u reči dodata je vrednost 128. Sama dužina tabele nije slučajno 256 bajtova. Ovakva dužina omogućuje indeksno adresiranje znakova u tabeli instrukcijama tipa LDA $A09E,X ili lda $A09E,Y

Službene reči

Službene reči, naravno, nije moguće menjati direktno u ROM memoriji, već je potrebno oformiti, negde na pogodnom mestu u RAM-u, potpuno novu tabelu, napisati nove rutine za rad sa tabelom i promeniti vektore na trećoj strani koji ukazuju na ove rutine. Tek posle ovoga računar će biti u stanju da razume nove reči kao što su

```
1410 ;      cmp #$ff
1420 ;      beq acit
1430 ;      bit $0f
1440 ;      bmi acit
1450 ;      sec
1460 ;      sbc #$7f
1470 ;      tax
1480 ;      sty $49
1490 ;      ldy #<ord-1  ;adresa
1491 ;         ;pocetka tabele slu-
1492 ;         ;zbenih reci bejzika
1500 ;      sty $fb      ;u pokazivac
1501 ;         ;za indirektno
1502 ;         ;adresiranje
1510 ;      ldy #>ord-1
1520 ;      sty $fc
1530 ;      ldy #0
1540 secr   dex
1550 ;      beq hela
1560 incr   inc $fb
1570 ;      bne lad
1580 ;      inc $fc
1590 lad    lda ($fb),y
1600 ;      bpl incr
1610 ;      bmi secr
1620 hela   iny
1630 ;      lda ($fb),y
1640 ;      bmi halt
1650 ;      jsr $ab47
1660 ;      bne hela
1670 halt   jmp $a6ef
1680 acit   jmp $a6f3
1690 ;
1700 ;
1710 ;rutina  za  uvodjenje bejzik
1712 ;u  tokene
1720 ;
1730 ;
1740 tokb   ldx $7a
1750 ;      ldy #$04
1760 ;      sty $0f
1770 a582   lda $0200,x
1780 ;      bpl a58e
1790 ;      cmp #$ff
1800 ;      beq a5c9
1810 ;      inx
1820 ;      bne a582
1830 a58e   cmp #$20
1840 ;      beq a5c9
1850 ;      sta $08
1860 ;      cmp #$22
1870 ;      beq a5ee
1880 ;      bit $0f
1890 ;      bvs a5c9
1900 ;      cmp #$3f
1910 ;      bne a5a4
1920 ;      lda #$99
1930 ;      bne a5c9
1940 a5a4   cmp #$30
1950 ;      bcc a5ac
1960 ;      cmp #$3c
1970 ;      bcc a5c9
1980 a5ac   sty $71
```

PIŠI, ČITAJ, IDI i tako dalje

Bejzik programe računar pamti u memoriji gotovo tačno onako kao što su i otkucani. Jedina razlika između izvornog zapisa programa i onog u memoriji je u „tokenizaciji" službenih reči. Tako će, na primer, reč „PROGRAM" biti zapamćena tačno onako kao što je otkucana, dok će reč „FORMAT" biti zapamćena u svega 4 bajta — u prvom bajtu će se nalaziti kod za službenu reč FOR, a u ostala tri tekst MAT. Ovakav način pamćenja i interpretacije bejzik programa u računaru C-64 obezbeđuju dve rutine.

Prva rutina se nalazi u okviru programa LIST i služi za prevođenje „tokena" u čitljive reči. Ova rutina nalazi se na adresi $A71A i u nju se ulazi preko vektora $0306. Rutina BTOK u asemblerskom progra-

```
1990 ;      ldy #<ord ;pocetna adresa
1991 ;         ;tabele sluzbenih
1992 ;         ;reci
2000 ;      sty $fb   ;u pokazivac za
2001 ;         ;indirektno adresiranje
2010 ;      ldy #>ord
2020 ;      sty $fc
2030 ;      ldy #$00
2040 ;      sty $0b
2050 ;      lda $fb
2060 ;      clc
2070 ;      adc #$ff        ;pokazivac =
2071 ;         ;pokazivac - 1
2080 ;      sta $fb
2090 ;      lda $fc
2100 ;      adc #$ff
2110 ;      sta $fc
2120 ;      stx $7a
2130 ;      dex
2140 a5ed   inc $fb         ;pokazivac =
2141 ;         ;pokazivac + 1
2150 ;      bne sekv
2160 ;      inc $fc
2170 sekv   inx
2180 a5e8   lda $0200,x
2190 ;      sec
2200 ;      sbc ($fb),y
2210 ;      beq a5bb
2220 ;      cmp #$80
2230 ;      bne a5f5
2240 ;      ora $0b
2250 a5c7   ldy $71
2260 a5c9   inx
2270 ;      iny
2280 ;      sta $01fb,y
2290 ;      lda $01fb,y
2300 ;      beq a609
2310 ;      sec
2320 ;      sbc #$3a
2330 ;      beq a5dc
2340 ;      cmp #$49
2350 ;      bne a5de
2360 a5dc   sta $0f
2370 a5de   sec
2380 ;      sbc #$55
2390 ;      bne a582
2400 ;      sta $08
2410 a5e5   lda $0200,x
2420 ;      beq a5c9
2430 ;      cmp $08
2440 ;      beq a5c9
2450 a5ee   iny
2460 ;      sta $01fb,y
2470 ;      inx
2480 ;      bne a5e5
2490 a5f5   ldx $7a
2500 ;      inc $0b
2510 a5f7   lda ($fb),y
2520 ;      php
2530 ;      inc $fb      ;pokazivac
2531 ;         ;pokazivac + 1
2540 ;      bne glaw
2550 ;      inc $fc
2560 glaw   plp
2570 ;      bpl a5f7
1390 ;
1400 btok   bpl exit
```

mu je gotovo doslovna kopija originalne. U njoj se koristi nova tabela simbola počev od adrese ORD. Promenjen je i način adresiranja tabele pošto dužina nove tabele prevazilazi 256 bajtova. Tabela se ovde adresira indirektno posredstvom memorijskih lokacija $FB i $FC. Ugradnjom ove rutine, računar će bilo koji bejzik program listati na srpskohrvatskom.

Druga rutina radi u okviru interpreterske petlje i služi za pronalaženje službenih reči u tekstu i njihovo pretvaranje u „tokene". Nalazi se na adresi $A57C i poziva se preko vektora $0304. Njoj u asemblerskom listingu odgovara rutina TOKB. Sve što je rečeno za prethodnu rutinu važi i za ovu. Po unošenju i inicijalizaciji ove rutine sve naredbe se moraju kucati

```
2580 ;      lda ($fb),y
2590 ;      bne a5b8
2600 ;      lda $0200,x
2610 ;      bpl a5c7
2620 a609   sta $01fd,y
2630 ;      dec $7b
2640 ;      lda #$ff
2650 ;      sta $7a
2660 ;      rts
2670 ;
2680 ;
2690 ;rutina za ispisivanje poruka
2691 ;bejzika i operativnog sistema
2700 ;
2710 ;
2720 tchd   pha
2730 ;      txa
2740 ;      pha
2750 ;      tax
2760  ;        lda  $186,c  ;adresa  sa
staka
2770 ;      cmp #$eb    ;test adrese na
2771 ;         ;skok  od  rutine  za
pri
2772     ;                ;kac     poruke
interpretera
2780  ;       bne  nije    ;na,sledeci
test
2790 ;      lda $107,c ;los bajt
2800 ;      cmp #$2f
2810  ;       bne  nije    ;na,sledeci
test
2820 ;      ldx #200
2830  mal     lda    $ab,x     ;racunja
adre-
2831  ;         ;na  poruka  pomocu
sbra
2832 ;         ;sa stare poruke
2840 ;      cmp #62
2850 ;      bne merd
2860 ;      lda nadl,c
2870 ;      cmp #63     ;high bajt lsal
2880 ;      bne merd
2890 ;      lda nad+1 x ;adresa nove
2891 ;         ;poruke u a,y
2900 ;      tay
2910 ;      lda nad,c
2920 ;       pha    ;u  privremeno  na
stek
2930 ;      txa
2940  ;       txa     ;postavi  sp   na
pstra
2941 ;         ;bnu vrednost
2950 ;      clc
2960 ;      adc #7
2970 ;      tax
2980 ;      pla   ;vrati a sa steka
2990 ;      txs   ;sp
3000  ;        jmp  $ab1e   ;na   staru
rutinu
3001 ;          ;za ispisivanje poruka
3010  merd    txs     ;postavi  x   na
stanscu
3020 ;      tsx    ;adresu
3030 ;      cpx #10     ;da li su
```

na srpskohrvatskom

... i službene poruke

Čim bejzik interpreter „zaključi" da se zbiva nešto što nije predviđeno, zaustavlja rad i izdaje poruku o grešci. Rutina za prijavljivanje grešaka nalazi se na adresi $A43A i poziva se odmah posle bejzik WARM START-a (vektor $0300). Same poruke grešaka nalaze se u

Bejzik računara „komodor 64" sigurno nije najbolja stvar koja se može videti — nezgrapan, sa nedovoljno instrukcija, ponekad sa čudnom sintaksom i mnogo „pešačkih" rešenja preko POKE instrukcije, on pre pripada istoriji računara. Međutim, kada se taj isti bejzik pogleda iznutra, ovaj utisak se unekoliko menja. Proizvođač bejzika i operativnog sistema Microsoft ostavio je veliki broj mogućnosti za izmenu i bejzika i operativnog sistema. Gotovo sve važnije rutine iz ROM-a pozivaju se preko vektora na trećoj strani RAM-a instrukcijama JMP ($03xx). Dovoljno je da korisnik napiše svoj program, koji će raditi umesto postojećeg u ROM-u, i da postavi odgovarajući vektor tako da ukazuje na početak nove rutine i računar već radi drukčije. Naravno, ovde treba biti veoma pažljiv, jer gotovo svaka greška dovodi do pada sistema i tada je jedini lek reset taster ili isključenje računara.

tabeli grešaka koja je organizovana isto kao i tabela službenih reči. Odmah iza ove tabele u ROM-u je smeštena lista početnih adresa za svaku moguću poruku. Pre ulaska u rutinu za ispisivanje greške u X registru mora se nalaziti redni broj greške. Rutina ERRO je kopija kraja WARM START rutine i početka rutine za prikaz grešaka. U njoj se jedino ze razliku od originalne rutine uz pomoć rednog broja greška biraju adrese novih poruka. Uvođenjem ove rutine računar će sve poruke prikazivati na našem jeziku.

Računar je sada na najboljem putu da ‚progovori' u potpunosti naš jezik. Sada se na engleskom prikazuju samo poruke bejzik interpretera kao što su READY, ERROR itd. i poruke operativnog sistema na primer PRESS PLAY ON TAPE.

Zamena ovih poruka nije tako jednostavna kao što su bile prethodne pošto rutine za njihov prikaz nisu rešene vektorskim pozivima. Jedna od ideja za rešavanje ovog problema bila je presresti neku vektorizovanu rutinu pa testirati da li je

```
3018 :          ;sve moguce adrese
3027 :          ;izmenjene
3040 :    bne del    ;ne na test
3050 nije  tax        ;test steka na
3051 :          ;poziv od rutine za
3052 :          ;ispis poruke opera
3053 :          ;tivnog sistema
3060 :    lda $100,x
3070      cmp #$37
3080      bne eksp   ;if ne nije u
3081           ;output
3090 :    lda $104,x
3100 :    cmp #$f1   ;high bajt
3110 :    bne sign   ;ne ... ;ispis
3120 :    tya
3130 :    ldx #0
3140 covj  cmp oduf,x ;pronalazenje
3141 :          ;ispravne poruke ko
3142 :          ;jeno u rastojan...
3143 :          ;isti ...
3144 :          ;... ispis ...
3150      beq ppme   ;ok ...
3160 :    inx
3170      bne covj   ;ne ponovo test
3180 ppme  lda nduf,x ;ispis nove
3181           ;poruke
3190      tay
3200 piem  lda nporuk,y
3210 :    cmp
3220      and #$7f
3230 :    jsr $ffd2
3240 :    iny
3250      pla
3260      bpl grad
3270      pla        ;sredi stek
3280 :    tax
3290 :    pla
3300 :    pla
3310 :    pla
3320 :    pla
3330 :    cli
3340 :    rts        ;povratak ;
3341 :          ;iz rutine 4 ...
3350 sign  pla        ;ne ... u output
3360      tax        ;sredi stek
3370      pla
3380 :    jmp $f1ca  ;na output
3390
3400 :
3410 ;nove rutine za ispis poruke
3411 ;o greskama
3420 :
3430 :
3440 erro  ...
3450 :    bmi lyes   ;greske postoji
3460 :    jmp $e474  ;ne na ready
3470 lyes  txa        ;broj greske ...
3480 :    asl a
3490 :    ta...
3500 :    lda ...    ;nadji adresu
3501 :          ;nove poruke
3510 :    sta $2...
3520 :    lda ...    ;high bajt
3530 :    jmp $a445  ;nazad u etapu
3540 :          ;rutinu ...
```

```
3532 :          ;poruke = greskama
3540 :
3550 :
3560 ;rutina za ...
3570 :
3580 :
3590 nnel  pha
3600 :    txa
3610 :    pha
3620 :    tya
3630 :    pha
3640 :    lda $474
3650 :    sta $...
3660 :    ldy #$50
3670 :    bpl $e5a
3680 :    jmp $e472
3690 4e5a  jsr $f502
3700 :    bne fe5e
3710 :       .byte  $ac,$48,$08;inp
;eeeee:
3720 4e5e  jsr $4abc
3730 :    jsr $44e1
3740 :    beq $e66
3750 :    jmp $4e72
3760 4e66  jsr $fd15
3770 :    jsr $44be
3780 :    jsr $e518
3790 :    lda  $ ...
;ponovo
3791 :          ;postavi bejzik vektore
3792 :          ;da ukazu na nove
3793 :          ;rutine
3800 :    lda #<chestte #32
3810 :    lda #nnel;ete # 18
3820 :    lda #nnel;ete # 19
3830 :       .byte  $ac,$07,$e0
;eeeee!
```

nju pozvale rutina za prikaz poruke. Ovde je očigledno najbolje koristiti rutinu OUTPUT. Ova rutina ispisuje jedan znak i pozivaju je sve rutine koje imaju potrebu za pisanjem pa sasvim sigurno i rutina za prikaz poruka. Rutina OUTPUT nalazi se na adresi $F1CA i poziva se preko vektora $0326.

Asemblerska rutine TCHO presreće rutinu OUTPUT, testira stek jer se na njemu mora nalaziti adresa za povratak u rutinu koja je pozvala OUTPUT. Testiranje se obavlja u dve faze. Prvo se testira da li je poziv uputio bejzik interpreter za ispis svoje poruke. Ukoliko je test pozitiven, ispisuje se nova poruka o grešci, a ako je negativen ispituje se da li je poziv uputila rutina za prikaz poruke operativnog sistema. Ako se i ovde dobije negativen odgovor, upravljanje se vraća ponovo na rutinu OUTPUT.

Kod ovakvog načina rada postoji još jedan dodatni problem. Ispis poruka interpretera obavlja se preko potprograma

[illegible]

za prikaz stringova koji lokacije $62 i $63 koristi za pamćenje početne adrese stringa. Uz to samu rutinu za prikaz greška može pozvati bilo koji deo programa. Kod poruka operativnog sistema stvar je još gora. Ispis se vrši preko posebne, naravno nevektorisane rutine koja zahteva u Y registru rastojanje početka poruke u bajtima od prvog bajta tabele poruka.

[illegible]

Jedini način za pronalaženje odgovarajuće poruke u novoj tabeli je upoređivanje pristiglih vrednosti sa mogućim. Zbog toga su uvedene tabele adresa starih i novih poruka interpretera OAO i NAD i stara i nova rastojanja za poruke operativnog sistema ODUF i NDUF.

```
Bejzik leader za programe bajzik YU
sa naredbama na e oznacen ja
tor asle, sa S cir a sa P ctrl?!

100 REM -
101 REM  BASIC Y  WRITEN BY RASA MA
102 REM  (C) COPYRIGHT BY ADP-CENTAR
103 REM    VRBAC, 26300 V TRG 1279
104 REM       TEL (013) 816 451
105 REM
106 POKE53281,1:PRINT"[illegible]SACEKAJTE 1 8 MINUTA MOLIM"
107 PRINT"[illegible]B E J Z I K   YU SE PUNI"
108 TI$="000000"
109 FORI=49152 TO 50993
110 PRINT [illegible]
111 READQ:POKEI Q:SU=SU+Q
112 NEXTI
113 IFSU<>[illegible]THEN[illegible]
114 PRINT [illegible]GRESKA U DATA NAREDBAMA
115 DATA [illegible]
116 DATA [illegible]
117 DATA [illegible]
118 DATA [illegible]
119 DATA [illegible]
120 DATA [illegible]
121 DATA [illegible]
122 DATA [illegible]
123 DATA [illegible]
124 DATA [illegible]
125 DATA [illegible]
126 DATA [illegible]
127 DATA [illegible]
128 DATA [illegible]
129 DATA [illegible]
130 DATA [illegible]
131 DATA [illegible]
132 DATA [illegible]
133 DATA [illegible]
134 DATA [illegible]
135 DATA [illegible]
136 DATA [illegible]
137 DATA [illegible]
138 DATA [illegible]
139 DATA [illegible]
140 DATA [illegible]
141 DATA [illegible]
142 DATA [illegible]
143 DATA [illegible]
144 DATA [illegible]
145 DATA [illegible]
146 DATA [illegible]
147 DATA [illegible]
148 DATA [illegible]
149 DATA [illegible]
[illegible]
```

Još malo komplikacija?

Vaš računar sada već sasvim tečno govori srpskohrvatski. No problemima nikad kraja. Pokušate li da neposlušan program zaustavite tipkama RUN/STOP i RESTORE sve poruke interpretera i operativnog sistema dobijeju se ponovo na engleskom. Jedino rešenje je promeniti i rutinu za NMI, koja se vektorski poziva preko $0318.

Ovaj posao obavlja rutina NNMI. Jedina razlika između originalne rutine i ove je u tome što NNMI po instaliranju vektora bejzika ponovo u njihove lokacije upisuje vrednosti koje ukazuju na nove rutine.

U prilogu 1 dat je asemblerski listing svih rutina neophodnih za rad i heks-damp svih pripadajućih tabela. Otkucajte program i prevedite ga pomoću PROFI ASS ili nekog drugog asemblera, uđite u neki od monitora, kucajte M C1F8 pa zatim pažljivo prekucajte heks-damp tabele. Program snimite iz monitora apsolutno na primer S „bejzik YU", 01, C000, C732, uzimajte ga sa kasete sa LOAD, a startujete sa SYS12*4096. Ako nemate ni asembler ni monitor otkucajte bejzik program iz priloga 2, prethodno ga snimite i startujte sa RUN i bejzik YU je pred vama. U prilogu 3 date su paralelno sve stare i nove naredbe kao i skraćenice za kucanje.

Startujte program bejzik YU i učitajte neki od programa koji ste ranije pisali. Kada izlistate program sve naredbe će biti automatski prevedene na naš jezik.

Startujte program sa RADI i on će, vrlo verovatno, raditi. Mogućnost za neispravan rad se krije u tokenizaciji teksta. Ako program ne radi proverite varijable — neka od njih sigurno sadrži službenu bejzik reč.

Tu nije kraj

Ovaj program se proteže od adrese $C000 do C731, pri čemu 501 bajt otpada na same rutine, a ostalo su tabele reči i poruka. Nemojte pokušavati da koristite ovaj program zajedno sa nekim od standardnih TURBO TAPE programa jer zauzimaju iste pozicije u memoriji. Jedan od najboljih načina za rešavanje ovog problema je premeštanje tabela pa i nekih delova rutina ispod ROM-a sa operativnim sistemom, ali to pokušajte sami da uredite.

Pošto se sve tabele nalaze u RAM-u uvođenje novih naredbi predstavlja banalan programski zadatak. Čitav ovaj posao svodi se praktično na pisanje programa za izvršenje nove naredbe a povezivanje sa već postojećim vrši se uključivanjem naziva naredbe u tabelu. Naravno ako se koristi ovaj metod potrebno je još dopisati


```
150 DATA [illegible]
[illegible DATA lines through 186]
```

tabela službenih reči u originalu i u bejzik YU sa skraćenicama za kucanje

END	KRAJ	kR
FOR	OD	-
NEXT	SLEDECI	sL
DATA	PODACI	pD
INPUT#	UNESI#	uN
INPUT	UNESI	-
DIM	NIZ	nI
READ	CITAJ	cI
LET	NEKA	nE
GOTO	IDIDO	iD
RUN	RADI	rA
IF	AKO	aK
RESTORE	VRATI	vR
GOSUB	POZOVI	poZ
RETURN	NAZAD	nA
REM	PORUKA	poR
STOP	STANI	sT
ON	PREMA	pR
WAIT	CEKAJ	cE
LOAD	PUNI	pU
SAVE	SNIMI	sN
VERIFY	PROVERI	prO
DEF	DEFINICIJA	dE
POKE	PAMTI	pA
PRINT#	PISI#	pI
PRINT	PISI	?
CONT	NASTAVI	naS
LIST	LISTAJ	lI
CLR	IZBRISI	iZ
CMD	IZLAZ	iZL
SYS	MASINAC	mA
OPEN	OTVORI	oT
CLOSE	ZATVORI	zA
GET	UZMI	uZM
NEW	BRISI	bR
TAB(	TAB(	tA

TO	DO	-
FN	FUNKCIJA	fU
SPC(	PRAZNO	prA
THEN	ONDA	oN
NOT	NIJE	nIJ
STEP	KORAK	kO
+	+	-
-	-	-
*	*	-
/	/	-
AND	I%	
OR	ILI%	IL
=	=	-
SGN	ZNAK	zN
INT	CELO	ceL
ABS	APSOLUTNO	aP
USR	KORISNO	korI
FRE	SLOBODNO	slO
POS	POS	
SQR	KOREN	korE
RND	SLUCAJNO	slU
LOG	LOG	lO
EXP	EXP	eX
COS	COS	cO
SIN	SIN	sI
TAN	TAN	-
ATN	ATN	aT
PEEK	BAJT	bA
LEN	DUZINA	dU
STR$	REC$	rE
VAL	VREDNOST	vrE
ASC	ASC	aS
CHR$	SLOVO$	slov
LEFT$	LEVO$	lE
RIGHT$	DESNO$	dE
MID$	SREDINA$	sR
GO	IDI	-

novu rutinu za izvršenje bejzik naredbi i priključiti joj tabelu adresa naredbi u ROM-u. Zgodna mogućnost je i šifrovanje programa — jednostavno izmenite redosled naredbi po nekom vašem ključu i u istom redosledu izmenite adrese. Posle ovoga program će raditi samo uz vaš bejzik. Na svakom drugom računaru sa normalnim bejzikom izazvaće konfuziju.

Isto tako svaku poruku možete jednostavno da izmenite a možete da dodate i nove poruke. Dodavanje novih poruka vrši se uključivanjem poruke u odgovarajuću tabelu, a njen prikaz vrši sa pozivom odgovarajuće stare rutine u ROM-u.

Radomir Nikolajev

„komodor" u mreži

Zbog načina na koji je izveden, RS232 interfejs na „komodoru 64" bi se mogao smatrati čistim dobitkom za vlasnike koji firmu nije koštao ni dinara. Većina „komodorista" smatra da je ovaj interfejs namenjen prvenstveno za povezivanje štampača. On, međutim, može dobiti i uzbudljiviju ulogu — njime se mogu povezati dva ili više računara u mrežu. Ova oblast je posebno interesantna zbog mogućnosti korišćenja jeftinog „komodora" kao periferne jedinice na kojoj se može obaviti mnogo „ozbiljnog" posla, a ono što prevazilazi njegove potencijale posla ti na obradu većem računaru.

Interfejs RS 232 je izveden na portu B čipa CIA #2 Operativni sistem koristeći tajmer A istog čipa generiše potreban tajming izlaznog signala dok se tajmerom B određuje tajming za ulazni signal i na osnovu toga formira primljeni bajt Ovakvo rešenje je jeftinije od upotrebe specijalizovanog čipa ali i postavlja određena ograničenja Da bi „komodor" detektovao nailazeći signal bez potrebe da se stalno vrti u petlji mora se koristiti interaptom koji će biti generisan pri svakom prelasku signala sa 1 na 0' na ulaznoj liniji za podatke Metod je jednostavan ali je nevolja u tome što isti ili sličan koncept koriste i tastatura kasetofon disk jedinica i printer Kako bi bilo koji zastoj zbog nekog drugog interapta ili sprečavanje mašinskom instrukcijom SEI poremetio tačnost očitavanje ulaznog signala i praktično onemogućio rad interfejsa morao je biti izabran nemaskirani interapt

U apsolutnom prioritetu

Tako je RS232 dobio potpuni prioritet u odnosu na ostale periferije pa se nepažljivim korišćenjem računar lako može zavrteti u mrtvoj petlji Ako je otvoren kanal za RS232 disk jedinica nikako neće uspeti da završi započeti posao učitavanja nekog programa! Isti interapt generiše i istovremeni pritisak na STOP i RESTORE tastere ali ovo ne može da izazove zabunu ako već želite da resetujete računar onda i nije važno šta će se desiti sa komunikacijom

Drugi problem koji proističe iz ovakvog rešenja je nemogućnost ostvarivanje svih brzina prenosa koji su definisani RS232 standardnom Pri otvaranju veze zadaje se željena brzina prenosa i na osnovu nje operativni sistem iz tablica u ROM-u uzima potrebne podatke za tajmer A Za brzinu od 300 bauda sadržaj tajmera će biti 310 (treba za svaki bit da odbroji 310 taktova sistemskog časovnika od 1 MHz) za 600 bauda je već samo 174, dok je za 1200 potrebno svega 105 otkucaja tako da veće brzine nije moguće ostvariti Zato je „komodor" na ovoj brzini stao iako je u „Reference Guide"-u dat način otvaranja veze za sva standardna brzina ona preko 1200 bauda nije moguće koristiti Po našem iskustvu i 1200 je već dovoljno kritična brzina pa je njeno korišćenje moguće samo u pažljivo napisanom mašinskom programu Za modem se uglavnom koristi 300 bauda dok je za „razgovor" dva komodora 600 bauda sasvim prihvatljiva brzina pa ovo ne predstavlja posebno ograničenje

Da bi isprobali komunikaciju preko RS232 interfejsa najjednostavnije će biti da povežete dva komodora Dovoljno su dva konektora za korisnički port i trožilni provodnik Vezu izvedite na sledeći način

SLIKA (1)

Ovakvim povezivanjem se ostvaruje najjednostavnija veza bez handšejkinga pri kojoj su moguće greške u komunikaciji ali može odlično da posluži za demonstraciju rada i razvoj softvera

Otvaranje veze

Veze za RS232 se otvara standardnom naredbom OPEN u sledećem obliku
OPEN logički broj veze 2 0 CHR$(kontrolni bajt)+CHR$(komandni bajt)

Polje za naziv pri otvaranju veze sa ostalim periferijama ovde je iskorišćeno za definisanje parametara komunikacija Ako ste komodor povezali na pokazan način u jedan od njih ukucajte sledeći kratak program

```
10 OPEN 30 2 0 CHR$(6)
20 GET#30 A$ IFA$<>  THENPRINTA$
30  GETB$ IFB$<> ' THEN  PRINT#30 B-
$ PRINTB$
40 GOTO20
```

Ovim programom može se voditi razgovor dva korisnika izabrana je brzina od 300 bauda i stop bit, dužina reči od 8 bitova bez provere parnosti Da bi se poveo razgovor treba startovati program ali se isti mora nalaziti i u drugom „komodoru" Primer je kratak pa ga je lako ukucati ali ćemo uraditi nešto bolje — zašto ga ne bismo poslali preko veze i tako u startu demonstrirali jednu od pogodnosti koju veza pruža Komodor nije predvideo SAVE i LOAD naredbe za RS232 pa će biti neophodno da u prijemnom računaru ipak napišemo dve programske linije

```
10 OPEN 7 2 0 CHR$(6)
20 GET#7 A$  PRINTA$   GOTO20
```

Sada u prvom u direktnom modu otkucajte sledeće

```
OPEN 30 2,0,CHR$(6)
CMD30 LIST
PRINT#30 CLOSE30
```

Na ekranu drugog računara će se pojaviti program iz prvog ali samo na ekranu Zato pritisnite STOP i kursorom pređite preko linija pritiskajući RETURN na svakoj Oba komodora će sada imati isti program pa možete početi sa razgovorom — sve što otkucate pojaviće se na vašem ekranu ali i na ekranu drugog komodora Ako istovremeno budete kucali slova će se mešati ali isto važi i pri običnom ljudskom razgovoru Ako je vaš sagovornik neprijatan pa vam ne da da dođete do reči već stalno kuca nešto možete mu lepo odgovoriti pritiskom na CLR taster Ovo će obrisati i vaš i njegov ekran ali i ilustrovati jednu lepu osobinu interfejsa ako je izabrana dužina reči od 8 bita mogu biti slati svi karakteri od 0 do 255 (CLR ima kod 147) Da ste izabrali dužinu reči od 7 bita osmi bit bi uvek bio 0 pa bi CLR taster samo pomerio kursor na početak ekrana Izbor dužina reči ne mora da vas brine — izaberite uvek 8 kada koristite vezu dva „komodora" Dužina od 7 bita predstavlja standard uglavnom pri komunikaciji sa većim računarima

Razvoj programa

Datom primeru nedostaje dosta toga da bi postao pravi program za komunikaciju Pošto je veza bila namenjena isključivo za razgovor nije bilo potrebno obezbeđenje od gubitka znakova pri prijemu jer je malo verovatno da možete da kucate brže od 300 bauda (približno 30 znakova u sekundi) Za tu brzinu prenosa i bejzik je dovoljno brz ali se pri većim brzinama ili manjim razmacima očitavanja ulaznih karaktera (sa GET) može desiti da računar jednostavno ne stigne sve da obradi Pošto se prijem kao i slanje, odvija pod nemaskiranim interaptom dakle nezavisno od trenutnog dešava-

nja u bejzik (ili mašinskom) programu potrebno je da se znakovi po prijemu (ili pre slanja) smeštaju u neki red tako da uvek budu na raspolaganju interapt rutini. Ovo je izvedeno tako što se u momentu otvaranja veze u vrhu memorijskog prostora dostupnog bejziku odvaja dva puta po 256 bajtova (ulazni i izlazni bafer). Kada se PRINT# pošaljete karakter na RS232 operativni sistem će ga samo proslediti u ovaj prostor i uvećati brojač znakova koji čekaju na slanje. Pri tome interapt rutina neprekidno uzima redom znakove iz ovog prostora i kontinualno obavlja slanje i smanjuje sadržaj brojača. Isto se odnosi i na prijem — u interaptu se znakovi primaju sa RS232 i redom smeštaju u bafer povećava brojač a GET# samo iz memorije uzima prvi primljeni i smanjuje sadržaj brojača. Ako se desi da interapt rutina prepuni bafer jer ne stižete dovoljnom brzinom da ih odatle pokupite ostatak će biti nepovratno izgubljen dok ne uzmete nekoliko znakova i tako otvorite prostor za nove. Još drastičnija stvar se dešava pri zatvaranju veze. Bez obzira da li je poslao sve znakove iz bafera ili ne „komodor" momentalno prekida komunikaciju i oslobađa bafere. Očigledno je da poznavanje stanja bafera pri iole većim brzinama postaje posebno važno. Način na koji operativni sistem odvaja bafere za ovu namenu detaljno je opisan u dodatku „Sve komodorove rutine" („Računari 6").

Informacija o stanju komunikacije dobija se iz standardne bejzik promenljive STATUS. Iako je ova bejzik varijabla ista i za sve ostale periferije, njen sadržaj se za RS232 formira na drugačiji način. Osnovna razlika je u tome što pri svakom testiranju njena vrednost dobija sadržaj 0 pa nije moguće na primer

IF (ST=0) OR (ST=8) THEN

jer nakon prvog logičnog izraza ST gubi originalni sadržaj pa drugi logički izraz nema smisao. Zbog toga je u ovakvim prilikama potrebno uvođenje nove varijable kojoj će biti dodeljen sadržaj STATUS-a a zatim njena vrednost testirana.

Svaki bit STATUS varijable ukazuje na neki tip greške u prenosu. Jedino bit 3 nema ovu namenu već označava da je prijemni bafer prazan i postavlja se nakon naredbe GET. Ostale greške i nije potrebno posebno testirati. Dovoljno je da utvrdite da ST nije jednako 0 ili 8 što znači da je bajt pogrešno primljen i da ga u najčešćem broju slučajeva jednostavno ignorišete.

Kako to kod „komodora" obično biva ovakav način nije dovoljan da pokrije sve moguće situacije. To se prvenstveno odnosi na slučaj kada treba da završimo razgovor i zatvorimo vezu a nemamo informaciju da li je interapt rutina uspela da pošalje čitavu poruku. Testiranjem sistemske varijable ENABL (673) čiji je bit 0 uvek resetovan kada je prenos završen može se rešiti i ovaj problem.

Poznavanje STATUS varijable i stanja bafera je dovoljno što se „komodora" tiče da se napiše korektan program za komunikaciju. Ostatak zavisi od toga koji će računar biti na drugoj strani linije.

Povezivanje sa većim računarima obično nameće samo jedan problem: konverziju ASCII kodova „komodora 64" na pravi ASCII kod. Uobičajeno je da se „komodor kod", kao što smo i mi u prethodnoj rečenici učinili, naziva ASCII iako s tim

```
10 REM ------------------------------------------------------------
20 REM              KONUNIKACIJR PREKO RS232 INTERFEJSA
30 REM ------------------------------------------------------------
90 CLR:POKE56,32:CLR                         :REM  REZERVISANJE NEMORIJE
100 POKE53280,0:POKE53281,0:PRINTCHR$(147):REM  IZBOR ONERRCIJE
110 PRINT"1. RAZGOVOR"
120 PRINT"2. PRENOS DATOTEKE"
130 PRINT"3. KRAJ RADA"
140 GETA$:IFA$=""GOTO140
150 ONVAL(A$)GOTO1000,2000,170
160 GOTO100
170 END
1000 REM
1001 REM          RAZGOVOR
1002 REM
1010 OPEN2,2,0,CHR$(6)+CHR$(0):GET#2,A$    :REM  INICIJRLIZRCIJA VEZE
1011 PRINTCHR$(147)"RAZGOVOR:":POKE204,0   :REM  KURSOR VIDJIV
1020 GETA$:IFA$=""GOTO1100                 :REM  OCITRVANJE TASTRTURE
1030 PRINT#2,A$;:PRINTCHR$(5)A$;           :REM  SLANJE KRRAKTERA
1040 IFA$=CHR$(95)GOTO6000                 :REM  PREKID VEZE (STRELICA LEVO)
1100 GET#2,B$:IFST<>0GOTO1020              :REM  PRIJEM, DA LI JE BAFER PRAZAN
1110 IFB$=CHR$(95)GOTO6000                 :REM  SAGOVORNIK PREKIDA RAZGOVOR
1120 IFB$=CHR$(128)GOTO3000                :REM  SAGOVORNIK SALJE DATOTEKU
1130 PRINTCHR$(28)B$;:GOTO1020             :REM  ISPIS PRIMLJENOG KARAKTERR
2000 REM
2001 REM  SLANJE DRTOTEKE
2002 REM
2010 GOSUB5000                             :REM  UPIS NAZIVR DATOTEKE
2020 OPEN4,8,3,N$:AD=8192                  :REM  UCITRVANJE U MEMORIJU
2030 GET#4,A$:IFST<>0GOTO2060
2040 POKEAD,ASC(A$+CHR$(0)):AD=AD+1
2050 GOTO2030
2060 POKE253,AD/256:POKE252,AD-256*PEEK(253):CLOSE4
2070 OPEN2,2,0,CHR$(6)+CHR$(0):GET#2,R$
2080 PRINT#2,CHR$(128);                    :REM  SRGOVORNIKU SE JRVLJR DA SLEDI
2090 FORF=1TO1000:GET#2,R$                 :REM  DATOTEKA, CEKA SE ODGOVOR DA JE
2100 IFA$=CHR$(128)GOTO2120                :REM  SPREMAN ZA PRIJEM
2110 NEXTF:GOTO6010                        :REM  ODGOVOR NIJE STIGAO, PREKID
2120 FORF=8192TOPEEK(252)+256*PEEK(253)-1 :REM  SLRNJE DRTOTEKE
2130 PRINT#2,CHR$(PEEK(F));:NEXTF
2140 GOTO6000                              :REM  KRAJ
3000 REM
3001 REM   PRIJEM DATOTEKE
3002 REM
3010 PRINT#2,CHR$(128);                    :REM  ODGOVOR, SPREMAN ZA PRIJEM
3020 PRINT"PRIJEM DATOTEKE:"
3030 AD=8192                               :REM  PRIJEM JE KONTINUALAN,
3040 FORF=1TO1000:GET#2,R$:IFST=0GOTO3130 :REM  PRVA PAUZR DUZR OD OKO 7 SEC.
3050 NEXTF                                 :REM  ZNACI DR JE PREDRJR ZRVRSENR
3060 POKE253,AD/256:POKE252,AD-256*PEEK(253)
3070 CLOSE2:GOSUB5000
3080 OPEN4,8,3,N$+",W"                     :REM  ZRPIS DATOTEKE NA DISK
3100 FORF=8192TOPEEK(252)+256*PEEK(253)-1
3110 PRINT#4,CHR$(PEEK(F));:NEXTF
3120 CLOSE4:GOTO100
3130 POKEAD,ASC(R$+CHR$(0)):AD=AD+1        :REM  KONTINUALAN PRIJEM I SMESTRNJE
3140 PRINTA$;:GOTO3040                     :REM  U MEMORIJU
5000 REM
5001 REM  UPIS NAZIVR
5002 REM
5010 PRINTCHR$(147)"NRZIV DATOTEKE: ";
5020 OPEN1,0:INPUT#1,N$:CLOSE1
5030 RETURN
6000 REM
6001 REM  KRRJ KOMUNIKRCIJE
6002 REM
6010 IF(PEEK(673)AND1)GOTO6010             :REM  IZLAZNI BAFER PRAZRN
6020 CLOSE2
6030 GOTO100
READY.
```

standardom ima malo veze. U drugom setu (velika i mala slova) ono što su u ASCII velika slova kod „komodora" su mala i obratno. Uz to kodovi od 96 do 127 zapravo i ne postoje, već su pomereni na 192 do 223 — iako će PRINT CHR$(97) dati na ekranu veliko slovo A, ako se GET ispitujemo tastaturu pa pritisnemo A+SHIFT dobićemo kod 193. Kod kontrolnih kodova je situacija komplikovanija jer postoji malo slaganja. Dovoljno je ipak znati da „komodorov" DEL kod (brisanje prethodnog znaka) u pravom ASCII-ju ima vrednost 8.

Objavljujamo i jednu malo bolju verziju programa za komunikaciju kojim se može

voditi razgovor korisnika dva „komodora 64" i ujedno razmenjivati datoteka sa diska Izborom opcija 1 možete voditi razgovor kao i sa prethodnim programom ali će sada pritisak na taster „strelica levo" bilo koga od dva korisnika prakinuti razgovor i oba vratiti na glavni meni Pranos datoteka se odvija izborom opcije 2 onog koji šalje i 1 onog koji prima Ostatak se pored zadavanja naziva odvija automatski Obratite pažnju kako je izveden protokol za ovu vrstu pranosa i kako je iskorišćen bajt 128 (proizvoljno izabrane vrednost) Ovo nije najsigurniji metod ali je dovoljno ilustrativan pa ga možete proširiti i time poboljšati program Zbog prostora i bolja preglednosti izostaviti smo standardne provera postojanja datoteka na disku itd

Komunikacija iz mašinca

Do sada je u „Računarima" bilo dosta reči o komunikaciji sa periferijama iz mašinskog jezika Za RS232 važi potpuno isto što i za ostala uređaje Standardna rutina operativnog sistema SETLFS SETNAM, OPEN CHKIN CHKOUT CHRIN CHROUT GETIN CLRCHN i CLOSE imaju iste namene pa praktično program koji je napisan za ispis na primar na printar, sada može — uz izmenu naziva datoteka koji nije bilo potrebno definisati i promenu prve adrese (2 umesto 4) — potpuno korektno da radi

Za uzimanja znakova sa veze ako su „komodori" spojeni na način koji smo ovde pokazali obavezna je upotreba rutina GETIN a ne CHRIN CHRIN rutina je namenjena za komunikaciju sa hendšejkingom i očekuje postojanja signala DSR, koji se u ovoj verijanti ne koristi pa je operativni sistem neće napuštati Tada jedino reset računara može pomoći

Komunikacija putem modema se odvija na potpuno identičan način kao i kod dva diraktno povezana „komodora" Uz modame se najčešća dobija i program koji je predviđen za ostvarivanja veze sa bilo kojim računaram pa je u njaga ugrađena sva potrabna konverzija kodova itd Zbog ovoga ih najčešće nije moguće koristiti za razmenu programa (mada ima i takvih) ali nema razloga da u tu svrhu ne koristite program koji smo dali Moguće je napraviti i sigurniju (mada sporiju) verziju koja bi koristila neku od metoda za proveru ispravnosti primljenog koda Na primer predajni računar bi mogao pri slanju računati i kontrolni zbir i na kraju poslati i njega da bi ga prijamni računar uporedio sa onim što je dobio računajući kontrolni zbir primljenih podataka Najsigurnija varijanta je provera bajt po bajt ali to znači dvostruko duža vrema za prenos programa što pri brzini od 300 bauda može preći tolerantnu granicu

Zoran Životić

Akcije

„Galaksijina" trakoteka

Ce biste nabavili naslove iz naše Biblioteke, jednostavno popunite narudžbenicu ili je, ako ne želite da oštetite svoje „Računare" prepišite na dopisnicu i pošaljite na adresu „Galaksija" (za Biblioteku programa), Bulevar vojvode Mišića 17, Beograd Verujemo da će naručeni programi biti isporučeni do kraja godine

Obzirom da će programi 1—5 moći da se nabave isključivo posredstvom našeg časopisa, svi poručioci će biti upisani u kompjutersku bazu podataka, što znači da će biti lično obavešteni o unapređenjima i novim verzijama programa (dokumentacija) i da će imati priliku da te verzije dobijaju ili besplatno ili uz minimalnu uplatu

Ekranski editor

1 Ekranski editor je namenjen svima koji pišu bejzik programe Omogućava efikasno ispravljanje softvera uz red sa 51 slovom u redu, dva kurzora, funkcijske tastere, prenumeraciju, zamrzavanje dela ekrana, različita skrolovanja i mnogo drugih stvari. Ovaj 100% mašinski program je potpuno kompatibilan sa raznim proširenjima za „spektrum" Pridodata je vrlo opsežna dokumentacija Autor Vladimir Kostić

Hiperbejzik

2. Hiperbejzik je unapređena verzija pobednika prošlogodišnjeg „Galaksijinog" konkursa Učitavši ga, bejziku vašeg „spektruma" dodajte tridesetak novih naredbi i, da stvar bude posebno lepa, i sami dobijete mogućnost da taj bejzik dalje proširujete bez ikakve potreba da se bavite asemblerom Pridodata je opsežna dokumentacija rađena prema standardima našeg konkursa Autor Fischer

Velika akcija

3. Velika akcija je prva kompleksna igra-avantura pisana potpuno na našem jeziku Kao diverzant treba da izvedete akciju u okupiranom gradu i prebegnete na slobodnu teritoriju Program je vrlo dopadljivo realizovan tehnika kompresije ekrana su omogućila da u memoriju stane veliki broj slika u koji, koje bivaju prikazana kad god dođete na neko mesto Pridodato je kratko uputstvo. Autor Aca Radovanović

Eatinglish 1

4 Eatinglish 1 program za učenje engleskog jezika kroz igru i zabavu, rađen je po principu koji su autori nazvali „All you wish" čime je program postao „nešto kao korpa sa igračkama iz koje se vadi šta se želi i kada se želi". Eatinglish ima preko 90 slika, rečnik od 390 reči kao i lekcije koje nastavnici engleskog mogu da modifikuju Pridodato je uputstvo rađeno prema standardima našeg konkursa Autori S Milekić i O Tanaskovski

Trodimanzionalna grafika

5. Trodimenzionalna grafika Jovana Skuljana — četrnaest kilobajta čistog mašinca — omogućuje grafičko predstavljanje elementarnih funkcija u tri dimenzije u aksonometriji i perspektivi Program je opremljen namenskim ekranskim editorom koji pojednostavljuje unošenje parametara i, da bi se dobilo na brzini prikaza, sopstvenim kalkulatorom

6 Modifikacija „spektrumovog" rom-a koja podržava rad na samo sa eprom-diskom nego i sa rom-ovima u sanci, detokenizuja tastaturu (naredbe se kucaju slovo po slovo) i uvodi nekoliko novih naredbi kao što je EDIT, i modifikacija GENS-a da može da radi iz rom-a

Servis za eprome

- ***Programiranje eproma sa modifikacijom „spektrumovog" rom-a*** 400
- ***Programiranje eproma sa modifikacijom GENS-a da radi iz rom-a*** 400
- ***Programiranje oba eproma (rom+gens)*** 600
- ***Programiranje eproma sa modifikacijom asamblera PROFI 64 i disasemblera da rade iz rom-a („komodor 64")*** 600

U eprom-servisu redakcije obavlja samo usluge programiranja To znači da je zajedno sa potvrdom o uplati na adresu redakcije potrebno poslati i odgovarajuće eprome (jedan ili dva eproma 27128 „spektrum" i jedan eprom 2764 ili 2732 za „komodor"). Programiranje će biti obavljeno najkasnije za deset dana Ako redakcija prekorači ovaj rok, programiraće eprom besplatno U cenu su uračunate usluge programiranja poštanskih troškova i neophodna modifikacija, a ne i samih programa Programi su pod kopirajtom i redakcija nema ni pravo ni nameru da ih distribuira Podrazumeva se da oni koji naručuju modifikacije već imaju osnovna verzije određenih programa.

NARUDŽBENICA

Ovim neopozivo naručujem pouzećem sledeće programe iz Galaksijine biblioteke

1 Ekranski editor	(spektrum)	900 din
2 Hiper bejzik	(spektrum)	700 din
3 Velika akcija	(spektrum)	400 din
4 Eatinglish	(spektrum)	500 din
5 Trodimenzionalna graf	(spektrum)	600 din
6 Modifikacija	spektrumovog rom-a	600 din

Takođe naručujem sledeće knjige

5. Sve spektrumove rutine	300 din
6 Sve komodorove rutine	300 din

Ime i prezime ______________________

Adresa ______________________

Mesto ______________________

Potpis ______________________

Odgovarajući iznos ću uplatiti poštaru prilikom preuzimanja programa

Programiranje u bejziku

matrice i nizovi

Nizovi

Numerički, ili brojni niz je skup promenljivih sa zajedničkim imenom kao recimo: **A(1), A(2), A(3),** i td. Da bi se preciziralo koliko članova neki niz sadrži, mora se obaviti njegovo dimenzionisanje, korišćenjem bejzik naredbe **DIM**. Na primer, ako izvršimo **DIM A(10)**, obavestićemo računar da uvodimo niz **A** od deset članova. U memoriji će se jednostavno, rezervisati potreban prostor za smeštanje deset promenljivih sa imenima **A(1)** ... **A(10)**, pri čemu je svaka od njih potpuno ravnopravna sa bilo kojom drugom bejzik promenljivom. Uostalom bejzik i ne poseduje naredbe za neposredan rad sa nizovima. Bilo bi divno kada bismo mogli da štampamo sve članove niza jednom jedinom naredbom **PRINT**. Neki jezici to mogu, a mi ćemo morati da tretiramo niz doslovno kao skup nezavisnih varijabli. Sa **PRINT A(1)**, recimo, štampali bismo samo prvi član.

Na prvi pogled čini se da ništa posebno ne dobijamo uvođenjem niza. Jednostavnije bi bilo koristiti obične promenljive A1, A2, A3 itd. ako ni zbog čega drugog, onda bar zbog kraćeg zapisa. Zato uzmimo jedan konkretan primer: zamislimo da vršimo neko merenje jačine struje u N vremenskih trenutaka. Podatke treba ubaciti u računar i zatim naći, recimo, srednju aritmetičku vrednost struje. Najprirodnije je dimenzionisati nizove **T(N)** i **I(N)**, gde je N ukupan broj merenja koji mora biti poznat unapred. Unošenje i ispisivanje podataka je u tom slučaju vrlo jednostavno.

```
1000 REM    ...
1010 LET A=
1020 FOR K=  TO 3
1030 LET A=A ...
1040 NEXT K
1050 LET A=SQR A
1060 RETURN
```

Srednja vrednost struje takođe ne predstavlja problem.

```
100 REM  SREDNJA VREDNOST
110 LET I=0
120 FOR K=1 TO N
130 LET I= +I(K)
140 NEXT K
150 LET I=I/N
160 STOP
```

Dakle korišćenjem nizova, programi postaju kratki i pregledni, što se nikako ne bi mogio izvesti sa varijablama T1, T2, T3 ..., I1, I2, I3 ... Nizove treba uvoditi uvek u situacijama kada radimo a većim brojem podataka iste prirode.

Vektori

Vektor je matematički objekat koji bi se mogao definisati kao uređeni skup brojeva, npr (3, 2, 5). Svaki od brojeva iz tog skupa predstavlja jednu k o m p o n e n t u vektora. Komponenti može biti jedna, dve, tri ili više, ali mi ćemo se prvo ograničiti na vektore sa manjim brojem komponenti, jer smo tada u stanju da ih i geometrijski predstavimo. Smatraćemo, zapravo, da je vektor neka orijentisana duž u prostoru, a da su komponente prosto projekcije te duži na ose koordinatnog sistema. Ili, ako početak vektora dovedemo u početak koordinatnog sistema, onda će krajnja tačka imati koordinate jednake komponentama tog vektora.



Sl. 1 Vektor

Treba samo imati u vidu da će se pri promeni koordinatnog sistema (recimo pri rotaciji) promeniti i komponente vektora **A**, mada njegov položaj i veličina ostaju neizmenjeni. Ali u fiksiranom koordinatnom sistemu, vektor A je potpuno i jednoznačno određen sa svoje tri komponente A_x, A_y i A_z. U programu ćemo razume se koristiti niz **A(1), A(2), A(3)**, dok bismo neki drugi vektor **B** prikazali sa **B(1), B(2), B(3)**.

Pokazaćemo sada kako se sve karakteristike jednog vektora mogu izračunati samo znajući njegove komponente. Prvo ćemo odrediti dužinu vektora ili njegov intenzitet. Rešenje dajemo u obliku potprograma što znači da sa **GO SUB 1000** dobijamo dužinu vektora **A**, ako smo unapred definisali niz **A(1), A(2), A(3)**.

```
3050 NEXT I
3060 GO ...
3070 ...
3080 FOR I=  TO N
3090 FOR K=1 TO N
3100 LET ... AND ... AND K=I
3110 NEXT K
3120 NEXT
3130 ...
3140 NEXT ...
3150 RETURN
```

Uglovi koje vektor **A** zaklapa sa osama koordinatnog sistema lako se nalaze kad je poznata dužina vektora. Sa **U(1)** označili smo ugao sa X-osom, sa **U(2)** ugao sa Y-osom, a sa **U(3)** ugao sa Z-osom.

```
5000 REM "SISTEM 3X3
5010 DATA +3  5 +1   + 3
5020 DATA +2 +4 +   +18
5030 DATA +4 +2  ...   6
5040 LET N=3
5050 DIM ...(N N)  DIM ...(N N)  DIM C(N)  DIM X(N)
5060 RESTORE 5010
5070 FOR I=1 TO N
5080 FOR J=1 TO N
5090 READ ...(I J)
5100 NEXT J
5110 READ ...(I)
5120 NEXT I
5130 GO SUB 3000
5140 PRINT  X= X(1)  Y= X(2)  Z= X(3)
5150 STOP
```

Zbir vektora **A** i **B** geometrijski se dobija njihovim nadovezivanjem, a algebarski sabiranjem odgovarajućih komponenti. Ako rezultujući vektor označimo sa **C**, onda možemo pisati potprogram:

```
1 ... REM ...
  FOR K=  TO 3
  ...
  ...
  NEXT K
  STOP
```

Vektori se mogu i množiti i to na dva načina. Mi ćemo ovde samo pomenuti tzv. skalarni proizvod, kada se kao rezultat dobija običan broj (skalar). Po definiciji skalarno množenja vektora vrši se tako što se pomnože dužine tih vektora i kosinus ugla među njima. Može se dokazati da se isti rezultat dobija ako se pomnože parovi odgovarajućih komponenti, a onda svi proizvodi saberu. Skalarni proizvod vektora **A** i **B** označili smo sa W.

```
... REM ... SA ...
... LET W=0
... FOR K=1 TO 3
... LET W=W+A(K)*B(K)
... NEXT K
... RETURN
```

Ugao između bilo koja dva vektora **A** i **B** lako se nalazi ako su poznate dužine vektora i odgovarajući skalarni proizvod

U=ACS (W/(A*B))

Treba jedino obratiti pažnju na činjenicu da funkcija **ACS** daje isključivo rezultate u intervalu između nule i 180 stepeni, a to ne more uvek odgovarati našim potrebama. Međutim, problem nije tako veliki. Ako nas umesto oštrog ugla **U**, koji nam daje funkcija **ACS**, interesuje onaj drugi tup, dobićemo ga sa **2*PI-U**.

Dimenzije

Sve što smo govorili u vezi sa vektorima u tri dimenzije, važi i za vektore u dve dimenzije — orijentisana duž u ravni. Dvodimenzioni vektori imaju samo dve komponente (projekcije na X i Y osu) a svi programi koje smo dali važe i u tom slučaju. Samo treba u **FOR-NEXT** petljama pisati 2 umesto 3 ili tretirati dvodimenzione vektore kao trodimenzione pri čemu je treća komponenta nula (nema projekcija na Z-osu).

Broj dimenzija može jednostavno, biti bilo koji prirodan broj: deset, sto, milion ... Zamislite, recimo, skalarni proizvod dva vektora od po hiljadu dimenzija. Zamislite uostalom skalarni proizvod dva vektora u četiri dimenzije. Šta je to četvorodimenzioni vektor? Svakako skup od četiri komponente, npr. (3, 7, 5, 2). Ali šta bi to bilo geometrijski? Potrebne su nam očigledno četiri međusobno normalne koordinatne ose a to je u našem trodimenzionom svetu nemoguće izvesti.

Standardni bejzik se, svakako, ne može ubrojati u jezike koji omogućuju korisniku struktuirano programiranje. Ipak, ako se nešto više pažnje posveti organizaciji podataka, može se i pomoću običnih FOR-NEXT petlji postići dobra struktura, preglednost i kratkoća programa. Pri tom mislimo pre svega, na nizove i matrice, a iz iskustva znamo da početnici uglavnom izbegavaju ovaj tip podataka, verovatno smatrajući da je to sve isuviše komplikovano. Pokušaćemo, zato, da približimo čitaocima pojam dimenzionisanih promenljivih, uz nekoliko primera njihove upotrebe u programima.

Računar se međutim ne bavi filozofijom i sa podjednako malo zanimanja za problem viših dimenzija računače uspešno sve operacije sa vektorima u hiljadu dimenzija baš kao da ih ima samo tri. Tako će recimo potprogram za nalaženje dužine vektora utvrditi da vektor (3, 7, 5, 2) ima dužinu 9.32738 ali ostaje van granica našeg poimanja kako i gde je postavljena ta duž.

Matrice

Kada smo rekli da se vektor može shvatiti kao uređeni skup brojeva nismo se dovoljno precizno izrazili mada smo podrazumevali uređenost u smislu niza brojevi slede jedan iza drugog. Međutim sasvim se lepo brojevi mogu poređati i u dve dimenzije

$$\begin{pmatrix} 4 & -3 & 7 & 2 \\ 6 & 4 & 4 & -1 \\ 8 & 0 & -4 & 3 \end{pmatrix}$$

Sl. 2 Matrica

Ovakva šema brojeva se u matematici zove matrica. Njen geometrijski smisao naravno ne možemo imati, jer se pojavljuje veliki broj komponenti (u našoj šemi čak 12). Međutim nekako se intuitivno nameće ideja da bi matrica trebalo da predstavlja uopštenje pojma vektora. Svaku vrstu i svaku kolonu matrice možemo shvatiti kao jedan vektor. Tada bi se moglo reći da se vektori sastoje od brojeva, a matrice od vektora. Ali tu je otprilike i kraj naše mašte. Matrice ćemo sabirati, množiti, transponovati, invertovati, a da pri tome praktično nećemo imati predstavu o tome šta radimo. A što se tiče računara, njemu će i dalje biti svejedno, što nam savršeno odgovara, jer nas ionako sada interesuje praktična primena matrica a ne filozofiranje na tu temu.

Kao najinteresantniji primer primene matrica uzećemo problem rešavanja sistema linearnih algebarskih jednačina. Sa tim se smo se svi sreli ako ne na drugom mestu onda bar u srednjoj školi, mada tamo niko verovatno nije radio sa matricama.

Sistemi jednačina

Počećemo sa jednostavnim sistemom od dve jednačine sa dve nepoznate

$$4x+5y=30$$
$$2x-3y=4$$

Zadatak je dakle pronaći takve vrednosti za x i y da date jednačine budu identički zadovoljene. Ima dosta načina da se pronađu te vrednosti za x i y. Ako imate iskustva sa matematikom onda rešenja možete naći za nekoliko sekundi samo gledajući u jednačine: drugu pomnožite sa dva, oduzmite od prve i tako eliminišete x. Bilo kako bilo, rešenje sistema je x=5 i y=2.

Dogovorićemo se sada da sistem linearnih jednačina zapisujemo u malo neuobičajenom obliku

$$\begin{pmatrix} 4 & 5 \\ 2 & -3 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 30 \\ 4 \end{pmatrix}$$

Slika 3 Sistem jednačina u matričnom obliku

Iskazano rečima to znači da proizvod matrice koeficijenata sistema i vektora nepoznatih daje vektor slobodnih članova

matrica × vektor = vektor

Istina, vektori su sada zapisani kao kolona (komponente su jedna iznad druge), ali to ne treba da nas buni. Naš cilj je da nađemo vektor (x, y).

Prvo ćemo matrici sistema dodeliti nešto što se zove determinanta

$$\begin{vmatrix} 4 & 5 \\ 2 & -3 \end{vmatrix}$$

Slika 4 Determinante

Determinanta za razliku od matrica nije nikakva šema sa neđokučivim smislom. Jednostavno one dve vertikalne crte sa strana označavaju *operaciju* koja se mora izvršiti nad brojevima u šemi, a rezultat te operacije će biti *najobičniji broj*. Pravilo je sledeće: *pomnožiti brojeve na glavnoj dijagonali i od toga oduzeti proizvod brojeva na sporednoj dijagonali*. Pri tome glavnom dijagonalom se naziva ona koja ide iz levog gornjeg ugla ka desnom donjem uglu matrice

$$D = \begin{vmatrix} 4 & 5 \\ 2 & -3 \end{vmatrix} = 4 \cdot (-3) - 5 \cdot 2 = -22$$

Slika 5 Rešavanje determinante

Determinanta našeg sistema jednačina iznosi $D=-22$

Treba izračunati još dve determinante D_x i D_y, koje se dobijaju kada se vektor slobodnih članova stavi u determinantu sistema umesto vektora koeficijenata uz *x* (prva kolona) a zatim uz *y* (druga kolona)

$$D_x = \begin{vmatrix} 30 & 5 \\ 4 & -3 \end{vmatrix} = -110 \quad , \quad D_y = \begin{vmatrix} 4 & 30 \\ 2 & 4 \end{vmatrix} = -44$$

Slika 6 Determinante promenljivih

Rešenje sistema se dobija po formulama

$$x=Dx/D=5$$
$$y=Dy/D=2$$

Savetujemo vam da ponovite ceo postupak za neki drugi sistem jednačina, recimo

$$8x-2y=10$$
$$4x+3y=33$$

Rešenja treba da budu $x=3$ $y=7$

Potpuno isti postupak važi i za sistem od *N* jednačina sa *N* nepoznatih $x1, x2, x3, \ldots, xN$. Opet treba izračunati determinantu sistema *D* i još ukupno *N* determinanti za nepoznate $D_{x1}, D_{x2}, D_{x3}, \ldots, D_{xN}$ koje se dobijaju tako što se vektor slobodnih članova stavlja na mesto odgovarajućih kolona u determinanti sistema. Rešenja su opet $x1=D_{x1}/D$, $x2=D_{x2}/D$ itd.

```
1200 REM "ZBIR VEKTORA"
1210 DIM C(3)
1220 FOR K=1 TO 3
1230 LET C(K)=A(K)+B(K)
1240 NEXT K
1250 RETURN
```

Nismo jedino objasnili kako se izračunava determinanta sa većim brojem vrsta i kolona. Tu takođe ima više načina a jedan je da se nekim transformacijama svi elementi determinante sa jedne strane glavne dijagonale (iznad ili ispod) dovedu na nulu. U tom slučaju je determinanta prosto jednaka proizvodu elemenata na glavnoj dijagonali. Prilikom transformisanja determinanta bilo koja vrsta se sme zameniti zbirom te iste vrste i neke druge, eventualno pomnožene proizvoljnim brojem (sabiranje vrsta se vrši kao sabiranje vektora). Isto naravno važi i za kolone.

```
1300 REM "SKALARNI PROIZVOD"
1310 LET W=0
1320 FOR K=1 TO 3
1330 LET W=W+A(K)*B(K)
1340 NEXT K
```

Kao prilog ovom tekstu naći ćete potprogram za računanje determinanta *D(N,N)* metodom koju smo ukratko i opisali. Rezultat će na povratku biti smešten u varijablu *DET*.

Takođe dajemo i kompletan potprogram za rešavanje sistema **N** jednačina sa **N** nepoznatih. Koeficijenti sistema se na ulazu moraju naći u matrici **F(N,N)** a slobodni članovi u nizu **G(N)**. Rešenje će biti dato nizom **X(N)**. Kao konkretan primer smo rešili sistem tri jednačine

$$3x-5y+z=3$$
$$2x+4y+6z=38$$
$$4x+7y-8z=-8$$

Rešenje je $x=3$ $y=2$ $z=4$

Napominjemo na kraju da rešavanje sistema linearnih jednačina nije baš tako

jednostavan posao, kako bi to možda mogio da izgleda na osnovu ovoga što smo govorili. Prvo, *ne mora* svaki sistem da ima rešenje. Ako se desi da determinanta sistema *D* bude nula, program koji smo dali će prijaviti grešku o aritmetičkom prekoračenju jer se vrši deljenje nulom pri izračunavanju samih nepoznatih. Dakle, program je

```
3000 REM        SISTEM: 'F(N N)*X(N)
3010 FOR I=1 TO N
3020 FOR J=1 TO N
3030 LET D(I J)=F(I I)
3040 NEXT I
```

predviđen samo za slučaj kada *D* nije nula. Drugi problem je tačnost sa kojom računar radi. Dovoljno je da su koeficijenti sistema neujednačeni po svojoj veličini, pa da ovim putem nikada ne dođete do rešenja.

Problema dakle ima, ali ćemo ih ostaviti onima koji se bave numeričkim metodama. Nama su sasvim dovoljni, bar za sada, neki od osnovnih principa i postupaka u programiranju.

Jovan Skuljan

Šta ima novo

Hitovi u senci

Protekla godina je bila veoma uspešna za programere. U prošlom broju „Računara" ukratko su prikazani najveći hitovi za spektrum, emsired i komodor. Međutim mnogo dobrih programa je ostalo u senci. Evo nekoliko takvih za spektrum.

Mnoge stanje igre su bile i dalje popularne među njima i Šebra Wulf, Knight Lore, Allien 8, Match point, Match day, Beach Head, Pyjamarama i mnoga druga. Početkom i sredinom 1985 pojavio se talas igara napravljenih krajem 1984. Pomenimo samo neke: B C Bill, Bruce Lee, Dark Star, Everyone's a Wally, Jet Set Willy 2, Sam Stoat, Spy vs Spy, Spy Hunter i još mnoge druge.

Od septembra pa na ovamo pojavilo se na desetine izuzetno vrednih igara: Night Shade, Hypersports, On the Run, The way of the Exploding fist, Frank B B, Herbert's Dummy Run, Glass... Iako je izgledalo da je to već previše, u poslednjih par meseci 1985 pojavilo se nekoliko igara koje imaju šansu da budu među najboljima u 1986. To su Marsport, Impossible mission, International basketball, Popay, Dynapila Dan, Scool Daze 2, Beach Head 2, Kung fu 2, Scooby Doo i još nekoliko njih.

Uslužnih programa je nešto manje ali su kvalitetniji: Megabasic, Artist, Beta basic 3.0, Designer's pencil, Dungeon Builder, Supercode 3.5 i drugi će se zasigurno koristiti još dugo.

Sve u svemu, uspešna godina za programere na spektrumu.

Ivan Nožinić

Majstorije na računaru zx spektrum

novi rom

GENS iz ROM-a

Na štampanoj ploči za proširenje ROM-a ima mesta za četiri eproma koje ćemo označiti sa ROM_0, ROM_1, ROM_2 i ROM_3. Pri tome je ROM_0 rezervisan za operativni sistem sa bejzik interpreterom, pošto se pri uključanju računara kontrola automatski prenosi na eprom u toj poziciji.

ROM_2 je predviđen za smeštanje asamblera, ali nije nikakav problem uz malu prepravku programa koja objavljujemo, da istu ulogu dobiju ROM_1 ili ROM_3.

Preporučujemo osim toga da jedna pozicija (recimo ROM_3) ostane rezervisana za originalni „spektrumov" ROM — prosto zato što se može desiti da neki komercijalni program ne podnosi novi hardver, pa je rešenje jedino u tome da se uključi originalni ROM. U tu svrhu iz bejzika samo treba izvršiti

OUT 239,3

Pri tome će sadržaj čitave memorije ostati naizmenjen. A povratak u ROM_0 može se izvršiti sa

OUT 239,0

Sadrzaj originalnog ROM-a je najjednostavnije prekopirati u jedan prazan eprom koji onda treba ubaciti u poziciju ROM_3. Međutim, na to mesto se sasvim lepo može staviti i sam ROM, pod uslovom da mu se na nožicu 27 direktno dovede signal MREQ, a prekine se veza koja spaja tu nožicu sa ostalim epromima.

ROM_2, kompletan DEVPAC

Kao što smo ranije već govorili *GENS3M* u toku rada koristi nekoliko potprograma iz osnovnog spektrumovog ROM-a. To malo komplikuje stvar oko organizacije memorijskog prostora, sve dok želimo da se potprogrami nalaze u istom ROM-u u kome se inače nalazi i *GENS*. Istina, mesta ima dovoljno ali se potprogrami moraju relocirati i skupiti u zajednički blok. Međutim, ako hoćemo da u isti ROM smestimo i disasembler *MONS3M*, pokazaće se da prostor od 16 kilobajta više nije dovoljan.

Daleko elegantnije rešenje bilo bi da se *GENS-u* omogući neposredno pozivanje potprograma iz osnovnog ROM-a. Time se ROM_2 oslobađa suvišnog tereta i u stanju je da prihvati čitav *DEVPAC*. Doduše, sam disasambler nije prilagođen radu iz ROM-a, ali on po svojoj prirodi treba da ostane relokatibilan kako bi bilo moguće analizirati pomoću njega proizvoljne segmente memorije. Kada korisnik to bude zahtevao *MONS* će biti iskopiran u RAM i tamo startovan.

Pitanje je, sada, kako osposobiti GENS da poziva potprograme direktno iz osnovnog ROM-a? Svakako ne dolazi u obzir da se ispred **CALL** naredbe prosto doda **OUT** instrukcija za prozivanje osnovnog ROM-a jer bi, odmah iza te **OUT** instrukcije procesor počeo da komunicira sa ROM-om koji je pozvan. Ukoliko tamo ne istim adresa-

```
*HI   T GENS3M2 AS
       ZX SPECTRUM

Copyright (C) HISOFT 1983,4
All rights reserved

Pass 1 errors  00

              1 *C-
5C3D    10 ERR_SP EQU   23613
5C49    20 E_PPC  EQU   23625
5C59    30 E_LINE EQU   23641
5C5D    40 CH_ADD EQU   23645
5C61    50 WORKSP EQU   23649
5C65    60 STKEND EQU   23653
5C92    70 MEMBOT EQU   23698
5CB2    80 RAMTOP EQU   23730
        90
0095   100 TOKTAB EQU   #0095
0D6B   110 CLS    EQU   #0D6B
0F2C   120 ED_ROM EQU   #0F2C
0F38   130 EDLOOP EQU   #0F38
0FA9   140 EDEDIT EQU   #0FA9
107F   150 ED_ERR EQU   #107F
12A2   160 MAIN   EQU   #12A2
12B4   170 ROM_M2 EQU   #12B4
1601   180 CHAN_O EQU   #1601
16B0   190 SETMIN EQU   #16B0
196E   200 L_ADDR EQU   #196E
1988   210 NXTONE EQU   #1988
19E8   220 RCLM_2 EQU   #19E8
1B17   230 LSCAN  EQU   #1B17
1BEE   240 CHCK_E EQU   #1BEE
1C82   250 EXPT_1 EQU   #1C82
1C8A   260 REP_C  EQU   #1C8A
1E99   270 F_INT2 EQU   #1E99
1E9F   280 REP_8  EQU   #1E9F
1EDA   290 REP_M  EQU   #1EDA
21B9   300 IN_ASS EQU   #21B9
24FB   310 SCAN   EQU   #24FB
2530   320 SYNT_Z EQU   #2530
       330
013D   340 STRT_G EQU   #013D
0181   350 WARM_G EQU   #0181
2000   360 MONS_A EQU   #2000
17C4   370 MONS_L EQU   #17C4
       380
3AAA   390        ORG   #3AAA
       400
3AAA   410 TEST_Z CP    #5C
3AAC   420        JR    NZ,BACK
3AAE   430        POP   HL
3AAF   440        POP   HL
3AB0   450        JP    REP_C
3AB3   460 BACK   LD    (CH_ADD),DE
3AB7   470        RET
3AB8   480 SETPPC CALL  GETNUM
3ABB   490        LD    (E_PPC),BC
3ABF   500        RET
3AC0   510 TOKENS DEC   HL
3AC1   520        LD    (CH_ADD),HL
3AC4   530        JR    TOKEN
3AC6   540 FIND_Q RST   #20
3AC7   550        CP    #0D
3AC9   560        RET   Z
3ACA   570        CP    #22
3ACC   580        JR    NZ,FIND_Q
3ACE   590 TOKEN  RST   #20
3ACF   600        CP    #0D
3AD1   610        RET   Z
3AD2   620        CP    #22
3AD4   630        JR    Z,FIND_Q
```

U tekstu „GENS iz ROM-a", objavljenom u „Računarima 9", pružili smo čitaocima ideju o tome kako da preprave svoj asembler i ubace ga u ROM računara. Međutim, za prepravljanje komercijalnih programa neophodno je izuzetno dobro poznavanje mašinskog jezika, a svesni smo činjenice da je za većinu korisnika daleko prihvatljivije rešenje kupovina gotovog programa, umesto disasembliranja nekoliko kilobajta čistog koda. Stoga, ove naše tekstove treba više shvatiti kao informaciju šta se sa GENS-om može uraditi, a manje kao uputstvo za samostalan rad. Uostalom, naš servis za eprome stoji na usluzi svim čitaocima. Projekat je, u protekla dva meseca, dosta usavršen. Razvijen je softver za rad sa novim asemblerom, i sada smo u prilici da konačno objavimo šta konkretno dobijaju čitaoci koji pošalju svoja eprome na programiranje.

```
[illegible] [illegible]    CP   [illegible]
3AD8  650         JR   Z,CONT
3ADA  660         CP   " "
3ADC  670         JR   Z,CONT
3ADE  680         CALL ALPHA
3AE1  690         JR   NC,TOKEN
3AE3  700         DEC  HL
3AE4  710         LD   A,(HL)
3AE5  720         CALL ALPHA
3AE8  730         JR   C,TOKEN
3AFA  740  CONT   LD   DE,TOKTAB
3AED  750         LD   BC,#A400
3AF0  760         LD   (MEMBOT),BC
3AF4  770  SEARCH CALL FIND
3AF7  780         JR   C,EXIT
3AF9  790         CALL COMP
3AFC  800         JR   C,SEARCH
3AFE  810         CALL CORR
3B01  820         JR   SEARCH
3B03  830  EXIT   LD   BC,(MEMBOT)
  J7  840         LD   A,C
3B08  850         AND  A
3B09  860         JR   Z,TOKEN
3B0B  870         LD   HL,(CH_ADD)
3B0E  880         DEC  HL
3B0F  890         LD   A,(HL)
3B10  900         INC  HL
3B11  910         CP   "*"
3B13  920         JR   NZ,PUT_TK
3B15  930         LD   A,B
3B16  940         CP   #CE
3B18  950         JR   NC,TOKEN
3B1A  960  PUT_TK LD   (HL),B
3B1B  970         LD   B,#00
3B1D  980         INC  HL
3B1E  990         DEC  BC
3B1F 1000         CALL RCLM
3B22 1010         JR   TOKEN
3B24 1020  FIND   LD   HL,(CH_ADD)
3B27 1030  MOVE   LD   A,(DE)
3B28 1040         INC  DE
3B29 1050         BIT  7,A
3B2B 1060         JR   Z,MOVE
3B2D 1070         INC  B
3B2E 1080         SCF
3B2F 1090         RET  Z
3B30 1100         LD   A,(DE)
3B31 1110         CP   (HL)
3B32 1120         RET  Z
3B33 1130         JR   MOVE
3B35 1140  COMP   LD   C,#01
3B37 1150  MATCH  INC  DE
3B38 1160         INC  HL
3B39 1170         INC  C
3B3A 1180         LD   A,(DE)
3B3B 1190         RES  7,A
3B3D 1200         CP   (HL)
3B3E 1210         SCF
3B3F 1220         RET  NZ
3B40 1230         LD   A,(DE)
3B41 1240         AND  #80
3B43 1250         JR   Z,MATCH
3B45 1260         INC  HL
3B46 1270         LD   A,(HL)
3B47 1280  ALPHA  CP   #41
3B49 1290         CCF
3B4A 1300         RET  NC
3B4B 1310         CP   #5B
3B4D 1320         RET
3B4E 1330  CORR   LD   A,C
3B4F 1340         CP   (IY+88)
3B52 1350         RET  C
3B53 1360         LD   (MEMBOT),BC
3B57 1370         RET
3B58 1380  RCLM   BIT  0 (IY+90)
3B5C 1390         JP   NZ,RCLM_2
3B5F 1400         LD   B,C
3B60 1410  SPCS   LD   (HL),#20
3B62 1420         INC  HL
3B63 1430         DJNZ SPCS
3B65 1440         DEC  HL
3B66 1450         RET
3B67 1460  PRGM_C RST  #10
3B6B 1470         LD   A,(HL)
3B69 1480         CP   #20
3B6B 1490         RET  NZ
3B6C 1500         INC  HL
3B6D 1510         RET
3B6E 1520  L_SCAN LD   HL,(E_LINE)
3B71 1530         SET  0,(IY+90)
3B75 1540         CALL TOKENS
3B78 1550         JP   LSCAN
3B7B 1560  INPUT  LD   HL,(WORKSP)
3B7E 1570         SET  0,(IY+90)
3B82 1580         CALL TOKENS
3B85 1590         JP   IN_ASS
3B88 1600  VAL    LD   HL,(CH_ADD)
3B8B 1610         PUSH HL
3B8C 1620         RES  0,(IY+90)
3B90 1630         CALL TOKENS
3B93 1640         POP  HL
3B94 1650         LD   (CH_ADD),HL
3B97 1660         JP   SCAN
3B9A 1670  GENS   CP   "#"
3B9C 1680         JR   Z,COLD_G
3B9E 1690         CALL CHCK_E
3BA1 1700         LD   HL,27001
3BA4 1710         LD   A,#ED
3BA6 1720         LD   B,#10
3BA8 1730  LP_G1  CP   (HL)
3BA9 1740         JR   NZ,COLDG1
3BAB 1750         INC  HL
3BAC 1760         DJNZ LP_G1
3BAE 1770         LD   BC,WARM_G
3BB1 1780         JR   RUN_G
3BB3 1790  COLD_G RST  #20
3BB4 1800         CALL CHCK_E
3BB7 1810  COLDG1 LD   DE,27000
3BBA 1820         LD   HL,(STKEND)
3BBD 1830         LD   BC,50
3BC0 1840         ADD  HL,BC
3BC1 1850         SBC  HL,DE
3BC3 1860         JP   NC,REP_M
3BC6 1870         EX   DE,HL
3BC7 1880         LD   (RAMTOP),HL
3BCA 1890         POP  DE
3BCB 1900         POP  BC
3BCC 1910         LD   (HL),#3E
3BCE 1920         DEC  HL
3BCF 1930         LD   SP,HL
3BD0 1940         PUSH BC
3BD1 1950         LD   (ERR_SP),SP
3BD5 1960         PUSH DE
3BD6 1970         INC  HL
3BD7 1980         INC  HL
3BD8 1990         LD   B,#10
3BDA 2000  LP_G2  LD   (HL),#ED
3BDC 2010         INC  HL
3BDD 2020         DJNZ LP_G2
3BDF 2030         LD   BC,STRT_G
3BE2 2040  RUN_G  LD   HL,RET_G
3BE5 2050         PUSH HL
3BE6 2060         PUSH BC
3BE7 2070         LD   A,#02
[illegible] 2080         OUT  (239),A
3BFB 2090         RET
3BEC 2100  COPY_M LD   E,C
3BED 2110         LD   D,B
3BEE 2120         LD   HL,MONS_A
3BF1 2130         LD   BC,MONS_L
3BF4 2140         PUSH DE
3BF5 2150         LD   A,#02
3BF7 2160         OUT  (239),A
3BF9 2170         LDIR
3BFB 2180         POP  BC
3BFC 2190         LD   HL,16
3BFF 2200         ADD  HL,BC
3C00 2210         LD   C,L
3C01 2220         LD   B,H
3C02 2230  RET_G  XOR  A
3C03 2240         OUT  (239),A
3C05 2250         RET
3C06 2260  MONS   CP   #0D
3C08 2270         JR   Z,WARM_M
3C0A 2280         CP   #3A
3C0C 2290         JR   Z,WARM_M
3C0E 2300         CALL EXPT_1
3C11 2310         CALL CNCK_E
3C14 2320         CALL F_INT2
3C17 2330  COLD_M CALL COPY_M
3C1A 2340         PUSH BC
3C1B 2350         RET
3C1C 2360  WARM_M CALL SYNT_Z
3C1F 2370         RET  Z
3C20 2380         LD   HL,(STKEND)
3C23 2390  LP_M1  LD   A,H
3C24 2400         OR   L
3C25 2410         JR   Z,T_COLD
3C27 2420         LD   A,(HL)
3C28 2430         INC  HL
3C29 2440         CP   #EF
3C2B 2450         JR   NZ,LP_M1
3C2D 2460         LD   B,#0F
3C2F 2470  LP_M2  LD   A,(HL)
3C30 2480         CP   #EF
3C32 2490         JR   NZ,LP_M1
3C34 2500         INC  HL
3C35 2510         DJNZ LP_M2
3C37 2520         LD   BC,29
3C3A 2530         ADD  HL,BC
3C3B 2540         LD   C,L
3C3C 2550         LD   B,H
3C3D 2560         PUSH BC
3C3E 2570         RET
3C3F 2580  T_COLD LD   BC,58000
3C42 2590         JR   COLD_M
3C44 2600  GETNUM CALL F_INT2
3C47 2610         LD   HL,9999
3C4A 2620         XOR  A
3C4B 2630         SBC  HL,BC
3C4D 2640         JP   C,REP_B
3C50 2650         RET
3C51 2660  EDIT   CALL GETNUM
3C54 2670         LD   (E_PPC),BC
3C58 2680         POP  HL
3C59 2690         POP  HL
3C5A 2700         LD   (IY+49),#02
3C5E 2710         CALL SETMIN
3C61 2720         XOR  A
3C62 2730         CALL CHAN_O
3C65 2740         LD   HL,ROM_M2
3C68 2750         PUSH HL
3C69 2760         LD   HL,(ERR_SP)
3C6C 2770         PUSH HL
3C6D 2780         LD   HL,ED_ERR
3C70 2790         PUSH HL
```

```
3C71  [illegible]        LD    ([illegible]),SP
3C75  2810               LD    HL,EDLOOP
3C78  2820               PUSH  HL
3C79  2830               JP    EDEDIT
3C7C  2840  DELETE       CALL  GETNUM
3C7F  2850               PUSH  BC
3C80  2860               CALL  SETPPC
3C83  2870               POP   DE
3C84  2880               LD    L,E
3C85  2890               LD    H,D
3C86  2900               AND   A
3C87  2910               SBC   HL,BC
3C89  2920               JP    C,REP_8
3C8C  2930               PUSH  DE
3C8D  2940               LD    L,C
3C8E  2950               LD    H,B
3C8F  2960               CALL  L_ADDR
3C92  2970               EX    (SP),HL
3C93  2980               CALL  L_ADDR
3C96  2990               LD    E,L
3C97  3000               LD    D,H
3C98  3010               CALL  Z,NXTONE
3C9B  3020               POP   HL
3C9C  3030               AND   A
3C9D  3040               EX    DE,HL
3C9E  3050               SBC   HL,DE
3CA0  3060               LD    C,L
3CA1  3070               LD    B,H
3CA2  3080               EX    DE,HL
3CA3  3090               CALL  NZ,RCLM_2
3CA6  3100               POP   HL
3CA7  3110               POP   HL
3CA8  3120               JP    MAIN
3CAB  3130  CAPS         SET   3,(IY+48)
3CAF  3140               JP    CLS
3CB2  3150  SET_DE       LD    DE,(CH_ADD)
3CB6  3160               LD    C,A
3CB7  3170               RST   #20
3CB8  3180               LD    A,C
3CB9  3190               RET
3CBA  3200  LET          CALL  TEST_Z
3CBD  3210               NOP
3CBE  3220               LD    A,#23
3CC0  3230               RET
3CC1  3240  SMALL        RST   #28
3CC2  3250               DEFB  #02
3CC3  3260               DEFB  #E2
3CC4  3270               DEFB  #38
3CC5  3280               RET
3CC6  3290  AD_ERD       INC   HL
3CC7  3300               LD    (HL),A
3CC8  3310               INC   HL
3CC9  3320               LD    (HL),E
3CCA  3330               INC   HL
3CCB  3340               LD    (HL),D
3CCC  3350               INC   HL
3CCD  3360               LD    (HL),0
3CCE  3370               DEC   HL
3CCF  3380               DEC   HL
3CD0  3390               DEC   HL
3CD1  3390               DEC   HL
3CD2  3400               SUB   #FF
3CD4  3410               OR    E
3CD5  3420               OR    D
3CD6  3430               POP   DE
3CD7  3440               RET   NZ
3CD8  3450               LD    (HL),#91
3CDA  3460               INC   HL
3CDB  3470               LD    (HL),#80
3CDD  3480               DEC   HL
3CDE  3490               RET
3CDF  3500  MSGADD       DEFB  #80
3CE0  3510               DEFM  "ZX "
3CE3  3520               DEFM  "Spectrum "
3CEC  3530               DEFM  "Extended "
3CF5  3540               DEFM  "BASIC Read"
3CFF  3550               DEFB  "y"+#80

Pass 2 errors: 00

Table used:   998  from  2000
```

ma ne postoji nastavak započetog programa doći će logično do kraha.

Najbezbednije je zato obavljati promenu ROM-ova pomoću programa koji radi iz RAM-a. Ako na primer *GENS* treba da pozove potprogram za očitavanje tastature on neće izvršiti **CALL&02BF**, već recimo **CALL KB_SUB**, gde je KB_SUB neki potprogram u RAM-u čija je uloga samo da prozove ROM_0, obavi **CALL&02BF**, ponovo prebaci ROM_2 i vrati se nazad u *GENS*

```
KB_SUB    SUB A
          OUT (239),A
          CALL&02BF
          LD A 2
          OUT (239),A
          RET
```

Isti princip se može primeniti na bilo koji program. Jedino treba voditi računa o tome da neki potprogrami imaju kao ulazni ili izlazni podatak sadržaj akumulatora, koji, u tom slučaju, treba da ostane sačuvan dok se izvršava prebacivanje ROM-ova. Na primer ROM_0 bi trebalo prozivati sa

```
          PUSH AF
          SUB A
          OUT (239) A
          POP AF
```

GENS3M dosta često koristi naredbu **RST&10** za štampanje ASCII znakova. To ne treba menjati ali se zato na adresi &0010 u istom ROM-u mora naći naredba **JP PR_SUB**, gde je PR_SUB odgovarajuća rutina u RAM-u koja poziva PRINT_A_2 iz osnovnog ROM-a na već opisan način.

Kompletan blok za pozivanje potprograma iz osnovnog ROM-a izgledao bi ovako

```
ROM_0     PUSH AF
          SUB A
          OUT (239),A
          POP AF
          RET
ROM_2     PUSH AF
          LD A,2
          OUT (239),A
          POP AF
          RET
CH_SUB    CALL ROM_0
          CALL&1601
          JR ROM_2
PR_SUB    CALL ROM_0
          CALL&15F2
          JR ROM_2
KB_SUB    CALL ROM_0
          CALL&02BF
          JR ROM_2
BP_SUB    CALL ROM_0
          CALL&03B5
          JR ROM_2
SA_SUB    CALL ROM_0
          CALL&04C2
          JR ROM_2
LD_SUB    CALL ROM_0
          CALL&0556
          JR ROM_2
```

Smatramo da posebna objašnjenja ovde nisu potrebna. Potprogrami ROM_0 i ROM_2 prozivaju odgovarajuće ROM-ove a CH_SUB PR_SUB KB_SUB BP_SUB SA_SUB, i LD_SUB su adrese koje *GENS* treba da poziva umesto CHAN_OPEN PRINT_A_2, KEYBOARD, BEEPER SA_BYTES i LD_BYTES. Na samom početku svog rada *GENS* će prvo formirati ovaj blok potprograma u RAM-u i tek onda nastaviti sa uobičajenim poslovima.

Da je nekim slučajem broj potprograma koje treba zvati iz osnovnog ROM-a bio mnogo veći rešenja koja smo upravo opisali ne bi se baš moglo svrstati u najracionalnija. Svaki novi potprogram tu zahteva dodatnih osam bajtova i tabela polako raste kao pantljičara.

Bilo bi sasvim moguće umesto toga napraviti jedan jedini program koji bi mogao da poziva bilo koju rutinu iz osnovnog ROM-a. Jedino se adresa rutine mora dostaviti tom našem univerzalnom programu u vidu ulaznog parametra. Na primer

```
CALL univerzalni program
DEFW adresa rutine
```

To međutim, znači da sada za svaki potprogram treba trošiti pet bajtova umesto tri koliko zahteva uobičajen poziv naredbom **CALL**.

Idealno rešenje sastoji se u upotrebi neke od **RST** instrukcija recimo **RST&30**, koja se retko koristi (*GENS* je ne koristi doslovno nikad). Time se poziv bilo kog potprograma može izvršiti sa samo tri bajta

```
RST&30
DEFW adresa rutine
```

Podrazumeva se naravno da na adresi &0030 stoji naredba **JP univerzalni program**. Ostaje još da se vidi kako će univerzalni program' prikupiti adresu rutine koja se poziva, ali o tome ćemo možda nekom drugom prilikom.

ROM_0: „novi spektrum''

Osnovni ROM je pretrpeo veći broj izmena kako bi se prilagodio novom hardveru i radu sa novim asemblerom. Osim toga smatrali smo da o istom trošku, treba ispraviti i sve poznate bagove originalnog spektrumovog' ROM-a. Uvedene su dalje četiri nove naredbe (umesto postojećih za mikrodrajv) a sve službene reči bejzika ukucavaju se se tastature slovo po slovo (ne postoji više kursor K')

Podrazumevaju se i sve izmene objavljene u tekstu ,ROM od sedam milja ali ćemo ovde navesti i njih radi potpunosti

Izmena 1
Pripremanje prostora u RAM-u za ROM_OS
&1235 LD HL,&5E53

Izmena 2
Rutina za inicijalizaciju

```
&04AA  CALL&0DEF
       LD A &C3
       LD (&5CB6) A
       LD HL &5CC0
       LD (&5CB7) HL
       LD A &C9
       LD (&5CB9) A
       CALL&386E
       RET
```

Izmena 3
Poziv rutine za inicijalizaciju
&126B CALL&04AA

Izmena 4
Obrada grešaka

```
a) &0008  CALL&5CB6
          LD HL (&5C5D)
          JR &005F
b) &005F  LD (&5C5F) HL
          JR &0053
```

Izmena 5
Postavljanje velikih slova (CAPS LOCK)
&1292 CALL&3CAB

Izmena 6
Aritmatička naredba bez LET
a) &1B41 CALL&3CB2
b) &1B46 CALL C &3CBA

Izmena 7
Štempanje programskog kursora
&187A CALL NZ &3B67

Izmena 8
Štampanje službenih reči (, tokana)
a) &0C17 JR &0C22
b) &0C38 RET

Izmena 9
Kursor K se nikad na štampa
```
&18FC NOP
      NOP
```

Izmena 10
Kursor K se ignoriša pri očitavanju tastetura
```
&0358 NOP
      NOP
```

Izmena 11
Tokenizacije programske linije
&12B4 CALL&3B6E

Izmena 12
Tokenizacije INPUT linija
&2159 CALL&3B7B

Izmena 13
Tokenizacije argumenta funkcija VAL
&35FD CALL&3B88

Izmena 14
Nove službene reči
```
&011F GENS
      MONS
      EDIT
      DELETE
```

Izmena 15.
Parametri novih naredbi
```
a) &1B06 DEFB&05
         DEFW&3C06  MONS
b) &1B0A DEFB&06
         DEFB&00
         DEFW&3C51, EDIT
c) &1B10 DEFB&06
         DEFB&00
         DEFW&3C7C, DELETE
d) &1B14 DEFB&05
         DEFW&3B9A  GENS
```

Izmena 16
Operacije poređenje i vrednovanja
e &353B RRCA
b &35E2 RRCA

Izmena 17
Štampanje realnih brojeva
&2E24 CALL&3CC1

Izmena 18
Funkcija INT
&3223 JR &323F

Izmena 19
Sabiranje celih brojeva
&3032 JP &3CC6

Izmena 20
Oparacija deljenje
&31FF JR Z &31DB

Izmena 21
Nova sistemska poruka
&1296 LD DE &3CDF

Izmena 22
Dperativni sistam ROM_OS počev od edresa &386E

Izmena 23
Nova naredbe i pomoćni potprogrami počev od edrese &3AAA

Nova poruka

Kade se računar uključi na ekranu se pojavljuja sistemske poruka ZX Spectrum Extanded BASIC Ready Kursor je automatski postavljen ne C i kucanje programa se more vršiti slovo po slovo Tek po pritisku tastera ENTER programske linija ća biti prevedene u standardni format gde sveka službene reč ima svoj jedinstveni kôd

Prilikom listanja programe linije se pojavljuju isključivo onako kako smo ih ukucali tj bez umetnutih praznih polje ispred ili iza službenih rači a programski kursor će se štampati ili između linijskog broja i prve neredba ili ako je prvi znek u liniji prazno polja štampanje kursore će se obaviti umesto tog praznog polje De bi sve ovo bilo jesnije najbolja je ukucati neki bejzik program pod kontrolom origineinog spektrumovog ROM-a a onde preći u ROM_0 i pogledeti listing

Osim ovih izmane vazanih za sam editor ispravljen je niz bagova o kojima je već toliko puta bilo rači i na stranicama Računera Funkcija **VAL, VAL$** i oparacija poređanja su sada ravnopravne sa ostalim oparacijama kalkulatora tako de se mogu slobodno pozvati iz lista koje prati instrukciju **RST&28** (izmene 16)

Izbegnuti su svi nesporazumi oko broja −65536 (izmana 18 i 19)

Vrlo grub bag (zaboravljene nula na računskom steku) u okviru rutine ze štampanja melih brojeva sada je otklonjen (izmene 17) i spektrum konačno ume de izračuna izraz tipe ,**1/2="+STR$0.5**

Poseban specijelitet novog ROM-a predstevlje mogućnost kucanja aritmetičkih naredbi bez obevezna službene reči **LET** Recimo neredba **A=2*PI** ća se izvršiti apsolutno identično naredbi **LET A=2*PI**, što je inače uobičejeno na svim računerima koji koriste bejzik

Nejzad korigovana je i omaška u programu za deljenje (izmana 20) a umesto neredbi za mikrodrajv (**CAT, FORMAT, MOVE** i **ERASE**) uvedena su četiri nove **GENS, MONS, EDIT** i **DELETE**, sa kodovime 207, 208 209 i 210 Opisećemo dejstvo sveke od njih

GENS

Naredba **GENS** poziva asembler *GENS3M RAMTOP* ća pri tome biti postavljen ne 27000 a iza te adrese će sa formirati radni prostor za potrebe asamblera

Ukoliko je bajzik blok prevelik tako da *RAMTOP* nija moguće spustiti tako nisko Imaćemo grešku M RAMTOP no good" Moraćemo tada de sami napravimo mesta primenom **CLEAR, DELETE** ili **NEW** neredbe

Sva ovo važi samo u slučeju kada se asambler poziva prvi put posle uključanja računara Međutim eko je program već bio jednom startovan, neredbe **GENS** ća obaviti običan vruć start

Kako to neredba **GENS** može znati da li je asembler bio ili nija bio startovan pre toga? Vrlo jednostavno prilikom inicijelizacija programe, odmah iznad *RAMTOP*-a se postavlja jedan blok od šesnaest bajtova koji su svi jednaki *&ED* Uloga tog bloka je samo de služi naradbi **GENS** kao znek raspoznavanja eko je blok prisutan, treba obaviti vruć start a ako ga neme obaviće sa hladan start sa inicijelizacijom

Ukoliko baz obzira ne bilo šta želimo izričito hladan start dodaćemo neredbi GENS znek # ' (CHR$ 35) **GENS #**

Povratek iz asemblera u bajzik (naredbom *B* u okviru *GENS* editora) vrši se na uobičajenu adresu za povratak iz svih bejzik neredbi (&1B76 STMT_RET) što znači da je neradbe **GENS** potpuno ravnopravna se bilo kojom drugom

MONS

Naredba **MONS** poziva disasemblar *MONS3M* Ako se program već nalazi u memoriji iznad bajzik bloka (prepoznaja se po bloku od šesnaest bajtova jednakih *&EF*) biće obavljen vruć start U protivnom *MONS3M* se kopira ne adresu 58000 i tamo startuje

Ukoliko želimo da startujemo disasembler na proizvoljnoj adresi *n* izvršićemo **MONS n** Paramatar *n* biće zaokružen ne najbliži ceo broj ali ako je van opsega (0 — 65535) imaćemo grešku B Integer out of range bez ikakvih promena u sedržaju memorije

Povratak iz disasemblera vrši se naporedno u glavnu izvršnu petlju bejzik editora što znači da ni jedne naredba iza **MONS** neće biti otkrivane

EDIT

Naredbe **EDIT n** postavlja progremski kursor na programsku liniju *n* i zatim preslikava tu liniju u prostor za editovanje na potpuno isti način keo da je bio pritisnut komandni taster *EDIT (SHIFT 1)* Parametar *n* se zaokružuje ne najbliži ceo broj Ako je van opsega (1 —9999) prijevljuja sa greška B Intagar out of range

Slično naredbi **MONS** povratak iz naredbe **EDIT** vrši se takođe u glavnu izvršnu patlju bajzik editora

DELETE

Naredbe **DELETE m, n** izbacuje iz listinge blok programskih linija počev od linije *m* pa zaključno sa linijom *n* Ako linije *m* na postoji podrazumeva se prve sledeća Ako linija *n* na postoji, podrazumeve se prethodna.

Peremetri *m* i *n* se zaokružuju ne najbliže cele brojeva a ako je bilo koji od njih ven opsega (1 —9999) ili je *m* veće od *n* imaćemo grešku B Integer out of range

Povratak se i ovda vrši direktno u bejzik editor

Jovan Skuljan

lagranžovi polinomi

Numerički metodi (2)

U prethodnom nastavku „Numeričkih metoda" uveden je zadatak interpolacije u kome na osnovu poznatih

$$x_0, x_1, x_2, \ldots, x_n \tag{1}$$

$$f(x_0)=y_0,\ f(x_1)=y_1, \ldots,\ f(x_n)=y_n \tag{2}$$

najpre treba odrediti neku funkciju F(x) koja zadovoljava uslov

$$f(x_0)=F(x_0),\ \ldots,\ f(x_n)=F(x_n) \tag{3}$$

a zatim traženu nepoznatu vrednost f(x) za zadato x izračunati kao F(x). Opisan je metod i dat program linearne interpolacije a uveden opšti pristup polinomijalnoj interpolaciji.

U ovom nastavku opisaćemo Lagranžov (Lagrange) metod i dati program koji ga realizuje.

Interpolacioni polinom

Određivanje interpolacionog polinoma koji će se koristiti za interpolaciju nepoznatih vrednosti f(x) može se pojednostaviti ako se on najpre zapiše u posebnoj pogodnoj formi. Cilj je neposredno dobiti vrednosti koeficijenata polinoma. Umesto da se polinom traži u standardnom obliku

$$F(x)=a_0+a_1x+a_2x^2+\ldots+a_nx^n \tag{4}$$

zapisaćemo ga u (Lagranžovom) obliku

$$F(x)=c_0\frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)}+c_1\frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)}+\ldots+c_n\frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})} \tag{5}$$

Lako je pokazati da je ovo zaista polinom stepena n. Brojilac svakog od sabiraka je polinom stepena n, imenioci su konstante pa ceo izraz F(x) kao zbir predstavlja polinom n-og stepena.

Postavimo sada uslov da polinom (5) prolazi kroz date tačke tj. uslov (3). Ovaj uslov se svodi na $f(x_i)=c_i$

$$f(x_0)=c_0,\quad f(x_1)=c_1,\quad \vdots\quad f(x_n)=c_n \tag{6}$$

pa su time koeficijenti Lagranžovog polinoma direktno određeni. Kada nađene vrednosti zamenimo u (5) dobija se definitivan oblik Lagranžovog interpolacionog polinoma

$$F(x)=f(x_0)\frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)}+f(x_1)\frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)}+\ldots+f(x_n)\frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\ldots(x_n-x_{n-1})} \tag{7}$$

Radi ilustracije na osnovu izvedene formule (7) formirajmo Lagranžov interpolacioni polinom koristeći sledeće vrednosti nezavisno promenljive x i odgovarajuće vrednosti funkcije f(x)

$$x_0=-1 \quad f(x_0)=1$$
$$x_1=0 \quad f(x_1)=1$$
$$x_2=1 \quad f(x_2)=1$$
$$x_3=2 \quad f(x_3)=-5$$

$$F(x)=1\frac{(x-0)(x-1)(x-2)}{(-1-0)(-1-1)(-1-2)}+1\frac{(x+1)(x-1)(x-2)}{(0+1)(0-1)(0-2)}+1\frac{(x+1)(x-0)(x-2)}{(1+1)(1-0)(1-2)}-5\frac{(x+1)(x-0)(x-1)}{(2+1)(2-0)(2-1)}$$
$$=-x^3+x+1$$

$$R_n(x)=f(x)-F(x) \tag{8}$$

Procena greške

Iznećemo sada formulu na osnovu koje se može proceniti greška koja se čini kada se tražena vrednost f(x) izračunava kao F(x). Ova greška se može nazvati greška interpolacione metode, radi toga što je sadržana u samoj metodi i ne treba je mešati sa greškom koja nastaje u računaru (ako se interpolirana vrednost određuje pomoću programa) radi računanja sa brojevima koji imaju ograničenu tačnost.

Obeležimo sa $R_n(x)$ razliku

$$R_n(x)=f(x)-F(x) \tag{8}$$

Ova razlika je jednaka nuli u svim tačkama

$$x_0, x_1, \ldots, x_n \tag{9}$$

a treba proceniti njenu vrednost u ostalim tačkama.

Ne ulazeći u detalje izvođenja (koje se bazira na konstrukciji specijalne funkcije i uzastopnoj primeni Rolove teoreme) navodimo formulu za grešku koja se dobija u obliku

$$R_n(x)=\frac{(x-x_0)(x-x_1)\ldots(x-x_n)}{(n+1)!}f^{(n+1)}(\xi) \tag{10}$$

gde je ksi neka tačka iz intervala

$$[x_0, x_n] \tag{11}$$

sve pod pretpostavkom da su tačke (9) uređene po rastućem redosledu a interpolaciona tačka x leži u intervalu datom sa (11).

Prokomentarišimo formulu za grešku.

Odmah se vidi da greška zavisi od x ij. od toga gde se u intervalu (11) nalazi tačka za koju treba interpolirati.

Pored ovoga, kako u formuli za grešku figuriše izvod reda n+1 funkcije f, to je jasno da greška zavisi od „tipa" funkcije i ili bolje rečeno od toga koliko je i pogodna za aproksimaciju polinomom. Kako polinom ne može da raste ili opada veoma brzo, to se ne može očekivati velika tačnost interpolacije za funkcije koje rastu ili opadaju mnogo brže kao što je recimo slučaj u blizini tačaka prekida. Nasuprot tome kod „sporih" funkcija ili funkcija sa visokom diferencijabilnošću mogu se očekivati tačniji rezultati.

Kada se ne poznaju izvodi funkcije f, onda se formula za grešku ne može primeniti. Tada se mogu izvršiti neka procena greške na osnovu tzv. konačnih razlika, ali o tome će biti reči u narednom nastavku.

Dodajmo još i to da se o položaju tačke ksi ne može ništa reći osim da pripada intervalu (11). Radi toga se formula za grešku često zamenjuje sa

$$|R_n(x)|\leq\frac{(x-x_0)(x-x_1)\ldots(x-x_n)}{(n+1)!}M_{n+1} \tag{12}$$

gde je M_{n+1} označen maksimum n+1-og izvoda na intervalu (11). Ovaj izraz za gornju granicu greška podrazumeva ne samo postojanje n+1-og izvoda funkcije f već i njegovu ograničenost.

U prethodnom nastavku je izostavljena formula za grešku u slučaju linearne interpolacije. Nju je sada moguće jednostavno dobiti kao specijalan slučaj formule (10). Dovoljno je u tom cilju primetiti da se linearna interpolacija može posmatrati kao polinomijalna interpolacija sa polinomom oblika

$$F(x)=a_0+a_1x \tag{13}$$

koji zadovoljava uslova

$$F(x_0)=f(x_0),\quad F(x_1)=f(x_1)$$

u tačkama x_0 i x_1 (slično bi bilo za bilo koji drugi podinterval $[x_i, x_{i+1}]$). U tom slučaju formula (10) postaje

$$R_1(x)=\frac{(x-x_0)(x-x_1)}{2}f''(\xi) \tag{14}$$

Navodimo bejzik program kojim se realizuje Lagrangeova interpolacija. Na početku se u dijalogu primaju ulazni podaci (vrednosti (1) i (2) i interpolacioni argument), omogućava njihove eventualne korekcije nakon unošenja, a zatim izdaje interpolirana vrednost. Na kraju se odgovorom na pitanje „Nova interpolacija ili kraj?" program može iskoristiti za računanje nove interpolirane vrednosti na osnovu istih podataka koji se ne moraju ponovo unositi.

Ilustrujmo jednim primerom rezultate programa i upotrebu formule za grešku. Neka treba interpolirati funkciju ln(x) za x=1 na osnovu sledećih datih vrednosti

$$x_0=0.8 \quad \ln(x_0)=-0.223143$$
$$x_1=0.9 \quad \ln(x_1)=-0.105360$$
$$x_2=1.1 \quad \ln(x_2)=0.095310$$
$$x_3=1.2 \quad \ln(x_3)=0.182321$$

Za x=1 na osnovu ovih ulaznih podataka program daje interpoliranu vrednost y=ln(1)=0.000103. Kako je tačna vrednost ln(1)=0 to je greška učinjena Lagranžovom interpolacijom

greška=0−0.000103=−0.000103

Procenimo sada grešku koristeći formulu (10). U ovom slučaju n=3 a potrebni četvrti izvod funkcije f se određuje kao

$$f'(x)=\frac{1}{x}$$

$$f''(x)=\frac{-1}{x^2}$$

Mnogi konkretni problemi koji se rešavaju na računarima imaju u svojoj osnovi zadatke numeričke matematik Numerički metodi poseduju veliku opštost i primenljivost, ali kako njihovo razumevanje i programiranje nije uvek sasvim jednostavno, to im upotreba, odnosno popularnost među vlasnicima mikroračunara, nije na nivou koji zaslužuju

Ova serija napisa ima za cilj da jednostavno, i samo u najvažnijim elementima, prikaže nekoliko metoda kojima se rešavaju neki osnovni zadaci numeričke matematike, kao i da ponudi programe koji su bazirani na tim metodama. Programi su rađeni na bejzik jeziku za „komodor 64", ali se jednostavno mogu preneti i na druge mikroračunare

```
100 REM ****************************************
105 REM *                                      *
110 REM *    LAGRANGEOVA    INTERPOLACIJA      *
115 REM *                                      *
120 REM ****************************************
125 REM * X-ULAZNA VREDNOST ARGUMENTA          *
130 REM *   INTERPOLACIJE                      *
135 REM * A-NIZ VREDNOSTI NEZAVISNO            *
140 REM *    PROMENLJIVE (ULAZ)                *
145 REM * V-ULAZNI NIZ VREDNOSTI FUNKCIJE      *
150 REM * N-BROJ DATIH TACAKA (ULAZ)           *
155 REM * Y-INTERPOLIRANA VREDNOST FUNKCIJE    *
160 REM *   (IZLAZ)                            *
165 REM *                                      *
170 REM ****************************************
175 PRINT CHR$(147)
180 PRINT" ************************************"
185 PRINT" * PROGRAM ZA INTERPOLACIJU FUNKCIJE *"
190 PRINT" *    JEDNE NEZAVISNO PROMENLJIVE    *"
195 PRINT" *                                   *"
200 PRINT" *     LAGRANGEOVA INTERPOLACIJA     *"
205 PRINT" ************************************"
210 NI=5
215 GOSUB530:REM IZBACIVANJE PRAZNIH REDOVA
220 INPUT" ( PRITISNITE RETURN DA NASTAVITE )";PR
225 PRINT CHR$(147)
230 DIM A(100),V(100)
235 REM ****************************************
240 REM *           ULAZNI PODACI              *
245 REM ****************************************
250 INPUT "UNESITE BROJ TACAKA";N
255 PRINT
260 PRINT"UNOSITE JEDAN ISPOD DRUGOG PAROVE"
265 PRINT"TACAKA  X(I),Y(I)"
270 PRINT
275 FOR L=1 TO N
280 PRINT"   X(";L;"),Y(";L;") =";
285 INPUT A(L),V(L)
290 NEXT L
295 PRINT
300 PRINT"AKO JE I-TI RED POGRESAN, UNESITE I"
305 INPUT"AKO NEMA GRESKE, UNESITE 0";M
310 PRINT
315 IF M=0 THEN 345:REM TACNO UNETO, DALJE
320 IF M>N THEN 295:REM VECE OD DIMENZIJE, PONOVO
325 PRINT"UNESITE X(";M;"),Y(";M;") =";
330 INPUT A(M),V(M):REM UNOSENJE ISPRAVKE
335 GOTO 295
340 PRINT
345 INPUT "UNESITE ARGUMENT INTERPOLACIJE";X
350 PRINT
355 PRINT
360 PRINT
365 PRINT"******** UNOS PODATAKA ZAVRSEN ********"
370 REM
375 REM ****************************************
380 REM *    VREDNOSTI LAGRANGEOVOG POLINOMA   *
385 REM ****************************************
390 Y=0
395 I=1
400 P=V(I):REM POCETAK CIKLUSA ZA RACUNANJE ZBIRA
405 J=1
410 IF J=I THEN 420:REM CIKLUS ZA PROIZVOD
415 P=P*(X-A(J))/(A(I)-A(J))
420 J=J+1
425 IF J<=N THEN 410
430 Y=Y+P
435 I=I+1
440 IF I<=N THEN 400
470 GOSUB 565:REM IZDAVANJE VREDNOSTI POLINOMA
475 REM ****************************************
480 REM *     KRAJ ILI NOVA INTERPOLACIJA      *
485 REM ****************************************
490 NI=5
495 GOSUB 530
500 INPUT"(1)-NOVA INTERPOLACIJA, (2)-KRAJ";W
505 IF W<>1 THEN END
510 GOTO 300
515 REM ****************************************
520 REM *     STAMPANJE NI PRAZNIH REDOVA      *
525 REM ****************************************
530 FOR M=1 TO NI
535 PRINT
540 NEXT M
545 RETURN
550 REM ****************************************
555 REM *  IZDAVANJE INTERPOLIRANE VREDNOSTI   *
560 REM ****************************************
565 PRINT
570 PRINT"INTERPOLIRANA VREDNOST FUNKCIJE"
575 PRINT"Y =";Y
580 PRINT
585 RETURN
```

$$f^{(3)}(x)=\frac{2}{x^3}$$

$$f^{(4)}(x)=\frac{-6}{x^4}$$

Dakle na osnovu (10) sledi

$$R_3(1)=\frac{(1-0.8)(1-0.9)(1-1.1)(1-1.2)}{4!}$$

$$=\frac{0.0004}{24}\cdot\frac{-6}{\xi^4}=\frac{-0.0001}{\xi^4}$$

gde za ksi znamo da leži negde u intervalu [0.8 1.2] mada tačnu vrednost za ksi ne možemo odrediti. Možemo međutim odrediti gornju i donju granicu za grešku. Kako je poslednji izraz za grešku monotone funkcije promenljive ksi to će njegove ekstremne vrednosti biti dostignute na granicama intervala [0.8 1.2]. Te vrednosti su respektivno −0.000244 i −0.000048. Dakle i greška za x=1 leži u intervalu [−0.000244 −0.000048]. Ovo se slaže sa gore određenom greškom od −0.000103 jer ova vrednost leži između upravo nađene gornje i donje granice za grešku.

U vezi sa greškom može se postaviti niz važnih i interesantnih pitanja. Kako redukovati grešku? Šta učiniti kada nam nisu poznati izvodi funkcije f? Da li je moguće interpolacijom dostići proizvoljnu unapred zadatu tačnost?

O ovim pitanjima kao i o Newtonovom (Njutnovom) interpolacionom polinomu biće reči u narednom nastavku.

Mr. Veljko Spasić

Radionica logičkih igara (2)

kamenčići na kompjuteru

Minimaks postupak, obzirom na mana koje smo izložili u prošlom broju, možete da koristita samo kod sasvim jednostavnih igara; za složenije igra nam je potrabno bolje oružje, što ne znači da ća nam upravo stečena znanje biti nekoriena. Ukoliko ste pra čitanja ove Škole u slobodnom vramenu razmišljali o programiranju neke logička igra, verovatno ate formulisali skup pravila koje neki igrač treba da sledi da bi pobedio Zar ta pravila ne bi bilo moguće ugraditi u program? Zašto da ne! Posmatrajmo, na primer, „kratić — kružič" igru čija pravila znaju sva deca. Program koji bi igrao koristeći sledeća pravila bi bio sasvim dobar jeste da ne bi mnogo pobedivao ali ne bi ni gubio

1 Ako postoji pobednički potez, odigraj ga

2 Ako protivnik ime pobednički potez, blokiraj ga

3 Ako je sredlšnje polje prazno, zauzmi ga

4 Odigraj bilo koji (RND) potez.

Rupice i kamenčići

Nije nam, međutim, cilj da pišemo programe koristeći neke hibride ljudskog razmišljanja i kompjuterske primene Lepše je (ali ne uvek i efektnije) razvijati makar rudimentarne oblike veštačke inteligencije pa ćemo se pozabaviti igrom „Kalah" (poznata i pod imenom „Awari") koja je da vas utešimo, komplikovanija nego „kratić—kružič"!

slika 14:

Na slici 14 je prikazana tabla za igru „Kalah" Učestvuju, jasno, igrači A i B. Ispred svakog od njih se nalazi šest rupa koje su obeležene brojevima 1—6 Sa desne strane svakoga od njih je specijalna rupa nazvana „kalah" U početku se u svakoj od 12 malih rupa nalazi po n kamenčića Igrač koji je na potezu uzima sve kamenčiće iz jedne od malih rupa koje se nalaze ispred njega i raspoređuje po jedan od njih u svaku od sledećih rupa u smeru obrnutom od kretanja kazaljke na satu Ukoliko je potrebno, neki od kamenčića ulaze u igračev „kalah", a neki mogu da preteknu čak i za protivničku stranu table, ali ne i za njegov „kalah" Postoje i dva dodatna pravila

Igrači naizmenično igraju po jedan potez osim kada zadnji od kamenčića koje neko raspoređuje završi u njegovom „kalahu" Taj igrač tad igra još jednom.

Ukoliko zadnji od kamenčića koje neko raspoređuje završi u praznoj rupi sa njegove strane, taj kamenčić se zajedno sa svim eventualnim kamenčićima u protivničkoj rupi se istim brojem seli u igračev „kalah"

Igra se završava kada neki od igrača ne može da odigra potez Tada se svi kamenčići iz malih rupa protivnika (ako ih ima) sele u njegov „kalah", pa pobeđuje onaj ko je u „kalahu" sakupio više kamenčića. Pravila ćete, naravno, najbolje uvežbati ako pokušate da igrate „kalah" sa nekim prijateljima ili, u nedostatku njega, sami sa sobom Čini se da je razumno odlučiti da se u svakoj od rupa u početku nađu 2 ili 3 kamenčića, igra sa 6 kamenčića u svakoj rupi je više nego teška!

Da je neko izložio pravila ove igre i rekao da ćete vi sastaviti program koji je dobro igra mislili biste da se šali! Videćete, međutim, da programirati „kalah" i slične igre nije naročito teško, treba samo ispravno pristupiti problemu i rešavati ga „u koracima preciziranja" U prvoj iteraciji ćemo sastaviti program koji ispituje svaki mogući potez i bira onaj posle koga se dolazi u najbolju moguću poziciju, docnije ćemo uzeti u obzir i ispitivanje odgovora protivnika kao i dalje povećanje „dubine razmišljanja

Da bi program mogao da izabere najbolji potez, treba mu pružiti mogućnost da ispita vrednosti pojedinih pozicija. Kako minimaks postupak ne dolazi u obzir (stablo „kalaha" je nezamislivo veće od stabla igre „poslednji pobeđuje?"), uvešćemo takozvanu statičku funkciju pozicije Jasno je da statičku funkciju pozicije možemo da izaberemo na mnogo raznih načina, odlučili smo da za početak izbrojimo sve kamenčića u rupama sa naše strane table i da im dodamo broj kamenčića u našem „kalahu" a da zatim od zbira oduzmemo broj kamenčića u otvorima koji pripadaju protivniku i u njegovom kalahu Na slici 15 je prikazana jedna moguća pozicija i svih pet poteza koje A može da odigra uz svaki od njih je upisana vrednost statičke funkcije pozicije što znači da će A odigrati srednji potez koji mu daje 6 kamenčića prednosti

slika 15:

Može i sasvim jednostavno

Pre nego što se prihvatimo pisanja programa moramo da se odlučimo za način na koji ćemo „pamtiti" neku poziciju Prvo što pada na pamet je koristiti dvodimenzionalnu matricu pit (7,2) u čije je elemente upisan broj kamenčića u svakoj od rupa Ukoliko hoćemo da pravimo nekakvu tabelu odluka, mogli bismo heksadekadno pakovati podatke pa ih „poukovati" u memoriju koju bismo na taj način štedeli. Nama se učinilo najjednostavnijim da dimenzionišemo niz pit (13) u čije elemente smeštamo broj kamenčića u svakoj od rupa. Iako ovaj pristup ne daje dovoljno uverljiv utisak o dvorednoj tabli sa slike 14, vrlo je pogodan za manipulisanje potezima. obzirom da je tabla za „kalah" kružna, raspoređivanje kamenčića u rupe može da se realizuje vrlo jednostavno koristeći funkciju MOD broj: =(broj+1) MOD 14 ća nam uvek dati broj sledeće rupe u koju treba da stavimo kamenčić Ovaj način „pamćenja" table je prikazan na slici 16

slika 16:

Što se pisanja programa tiča potrebna nam je najpre procedura koja će ispisivati na ekranu trenutnu poziciju Obzirom da ne želimo mnogo da se bavimo grafikom koja i onako znatno varira od računara do računara, jednostavno ćemo ispisati broj kamenčića u svakoj od rupa tako da se na ekranu dobije utisak o dvodimenzionalnoj tabli, ukoliko ste ljubitelj grafike pokušajte da uradite nešto bolje Potrebna nam je osim toga, procedura koja će inicijalizovati igru (tj staviti po nekoliko kamenčića u svaku rupu i isprazniti „kalahe") kao i procedura koja ća testirati kraj igre i proglasiti pobednika Ovi potprogrami su prikazani na slici 17 i nije im potreban nikakav naročit komentar

slika 17

[illegible]

U „Računarima 2" i „Računarima 3" smo objavili škole akcionih i avanturističkih igara koje su, nadamo se pomogli domaćim vlasnicima računara da se uhvate u koštac sa na tako jednostavnim problemom programiranja igara. U ovim i sledećim „Računarima" ćemo čitavu stvar dopuniti — videćemo kako se pišu logička igra. U školi igara avantura, nismo, jasno, pisali program ranga „Hobbita", pa ni sada nećemo pisati program za šah. Ipak, praćenja ovog napisa treba da pruži uvid u osnove tzv. veštačka inteligencije da vam, ako ništa drugo, bar pruži priliku da razumete kako otprilika rade programi koji igraju šah i slične igre. Pod pojmom „slična" podrazumevamo igre dva protivnika koji imaju podjednaka šanse za pobedu na tabli koja nije skrivena (igre tipa „Podmornice" ćemo programirati nakom drugom prilikom) i u kojima sreća tipa „slučajan broj" nema nikakvu ulogu.

Potrebna su nam dalja procedura koja će sa tastatura primati poteza igrača i odigravati poteza kompjutarovog protivnika koga smo nazvali igrač B. Kao i u prethodnom primaru sastavićemo odvojenu proceduru koja ća povlačiti poteza igrača A (kompjutara) kako bismo izbegli žongliranja sa promanljivom potez. Ove procedura su prikazana na slici 18, i dalja samo priman elemantarne programeraka veština.

Došlo je najzad vrama da sastavimo tunkciju koja generiše najbolji mogući potez — srca našeg programa. Obzirom da ćemo se za sada zadovoljiti samo razmatranjem svake moguće pozicija i njene statička funkcija procedura sa slika 19 su ralativno jednostavna. tryAmove povlači potez zatim se pomoću staticval nalazi vrednost dobijana pozicija a zatim moveback vraća sva u prathodno stanja. Na kraju bestmove bira potez koji ja dao najveću statičku vrednost. Interesantno je da smo proceduri moveback dali i četvrti paramatar, oznaku igrača koji je tranutno na potezu i čiji potez treba vratiti. Ovo ća nam biti korisno tek kada budamo razmatrali odgovora igrača B na poteze računara, za sada moveback vraća samo poteza igrača A.

slika 19

Skup procedura koja smo definisali zajedno sa glavnim programom sa slika 19 ispravno ali vrlo slabo igra „kalah". Ukoliko smatrate da niste dobro razumeli dosadašnji deo teksta, otkucajte program i pokušajte da osetite, kako diše. Ukoliko smatrate da ste ga dobro razumeli, pokušajte da unapredite igru ne dirajući mnogo u program. Najviše ćete postići unapređujući staticval. Ukoliko se, na primer, nakupi mnogo kamenčića otvoru broj 6 nekog od igrača, oni će vrlo skoro morati da budu premeštani na stranu protivnika, pa predstavljaju pre slabost nego snagu. Kredit koji igrač dobija za kamenčiće u rupama koje su dalje od protivničke strane bi trebalo povećati, trebalo bi osim toga više bodovati kamenčiće u „kalahu" i prazna otvore koji će eventualno poslužiti za zarobljavanja protivničkih kamenčića. Ovakvim izmenama ćete možda uspeti da unapredite rad programa, ali će prava poboljšanja moći da nastanu tek njegovim proširivanjem.

Ka boljem „kalahu"

Iako smo već rekli da korišćenje minimaks kriterijuma ne dolazi u obzir zbog veličine stabla igre „kalah", možemo da pristanemo na kompromis: ispitivaćemo nekoliko poteza u dubinu primenjujući minimaks kriterijum na vrednosti pozicija koje procenjujemo statičkom funkcijom. Što se budemo odlučili za veću dubinu program će bolje igrati, ali će utrošak memorije (svaka rekurzija odnosi ne baš mali broj bajtova) i procesorskog vremena biti neprijatniji. Na slici 20 je prikazan deo stabla hipotetične igre koju ispitujemo primenom mešavine minimaks postupka i statičke funkcije pozicija. Početnoj poziciji je vidimo pridružan broj 3.

slika 20

Funkcija koja primenjuje ovaj postupak je u suštini sasvim slična funkciji koju smo sastavili pripremajući program za igru „poslednji dobija?". Na slici 21 su prikazane procedure minval (poziva se kada je na potezu B) i maxval (poziva se kada je na potezu A) kao i nova verzija funkcije tryAmove. Uveli smo i globalnu promenljivu maxdepth koja treba da odredi maksimalnu dubinu ispitivanja pozicija, na primer maxdepth =3.

Nove procedure koje realizuju minimaks postupak zajedno sa odgovarajućim potprogramima testAmove i testBmove vam možda izgledaju prilično komplikovano ali

bi trebalo da budete u stanju da primetite da se one u suštini ne razlikuju od onih koje smo pisali ranije. Pozivanje minval testira argument depth i, ako nije dostignut maxdepth, proba da rasporedi kamenčiće iz svake od rupa (testBmove) i preračuna vrednost dobijene pozicije rekurzivno pozivajući minval odnosno maxval uz povećanje depth za 1. Svaki isprobani potez se odmah poništava koristeći moveback.

```
[illegible]
```

Žalosno je čuti da procedure koje smo teškom mukom napisali ne možemo ni za šta da iskoristimo! Paskal je, naime, jezik koji su pisali matematičari a ne inženjeri što znači da je izvanredan za sve — osim za ono što vam je upravo potrebno! U našem slučaju se pokazalo da je nemoguće napisati proceduru koja poziva neku drugu proceduru, pri čemu ta druga procedura rekurzivno ponovo poziva prvu. Ako pažljivije pogledate videćete da se minval, maxval, testAmove i testBmove intenzivno pozivaju među sobom. Zato smo morali da se poslužimo trikovima i sastavimo konačan program za Kaleh sa slike 22; verujte nam na reč da u njemu nema ništa osim onoga što smo ranije napisali i da su neki delovi koda morali da budu ponavljani samo da bi se zaobišle mane paskala (da slededći put pređemo na Adu?)

```
[illegible]
```

```
[illegible]
```

Isprobali smo oveko dobijeni program na BBC B koji je poznat kao prilično brz kompjuter i ustanovili da postavljanje maxdepth =3 zahteva desetak sekundi „razmišljenja" za svaki potez, dok maxdepth 4 produžava ovo bar po pet puta. Pokazalo se međutim, da i maxdepth =2 daje sasvim pristojnu igru računara koga ćamo pobeđivati tek kada steknemo malo više iskustva (ako poželite da povećate brzinu rada programa prevodeći ga na asembler poslušajte prijateljski savet i odustanite od toga!)

Kalah bez paskala

Pošto smo napisali kostur programa nije ga teško poboljšavati i modifikovati. Najperspektivnija mogućnost je selektivnost pri odluci o prekidu dalje analize pozicije: umesto da se zaustevimo kada dostignemo maxdepth možemo da predvidimo da se ispitivanje prekida kada je dostignuta maksimalna dubina i kada je sekenca uzimanja završena (često se dešava da naša verzija programa ne primeti da će posle spektakularnog skoka skora doći još spektakularniji pad jer će protivnik u sledećem potezu uzeti ogroman broj kamenčića). Modifikacija je više nego jednostavna: treba promeniti minval i maxval tako da konačni uslovi glase

IF (depth>=maxdepth) and (caprute <=0) then min =staticval

Iako smo na ovaj način stigli do kraja drugog dela naše Škole logičkih igara nismo ni iz daleka rekli sve što bi moglo da se kaže na nivou za koji smo se opredelili. Želeli smo, naime, da pokažemo da se logičke igre mogu napisati bez mnogo logičkog razmišljanja: naš program za Kalah bi uz vrlo male prepravke pobeđivao u bilo kojoj drugoj logičkoj igri što znači da nije teško simulirati inteligenciju korišćenjem isključivo procesorskih moći računara. Dok čekate naš sledeći nastavak u kome ćemo se pozabaviti tehnikama poznatim kao tree pruning (već ćemo se setiti kako to da prevedemo) zabavite se sledećim problemom — nagradićemo najbolje rešenje.

Možete li da programirate „Kalah" na bejziku? A na „galeksijinom" bejziku? Naš Katalog programa je nekada imao program za igru „Awari" pisan za TRS 80 koji ne omogućava rekurziju: program je čak učio u toku igre!

Dejan Ristanović

hiperboličke funkcije

Matematički softver

To može i bolje (3)

Korisnici računara veruju da se vrednosti hiperboličkih funkcija jednostavno računaju svođenjem na pouzdanu firminu eksponencijalnu funkciju EXP. Programeri znaju da je to verovanje dvostruko pogrešno: niti se računanje vrednosti hiperboličkih funkcija može svesti na eksponencijalnu funkciju, niti je firmin program EXP pouzdan. Prof. dr Dušan Slavić nastavlja seriju „To može i bolje", koja sadrži detalja njegovog dve decenije dugog istraživanja u oblasti računarskih algoritama.

Program za izračunavanje vrednosti računarske funkcije je valjan ako za svaku vrednost argumenta daje vrednost funkcije tačnu do poslednjeg bita mantise. Ovde pretpostavljamo da je program za izračunavanje vrednosti eksponencijalne funkcije valjan. U Računarima 10 (str 54—56) dat je takav program. Uz pretpostavku da je EXP valjan program istraživanje pokazuje da se vrednosti hiperboličkih funkcija

(1) sh(x)=(exp(x)−exp(−x))/2
(2) ch(x)=(exp(x)+exp(−x))/2
(3) th(x)=(exp(2x)−1)/(exp(2x)+1)

ne mogu dobiti direktnom primenom formula (1) (2) i (3). To je veliko iznenađenje za korisnike računara koji misle da za programiranje nije potrebno poznavati numeričku matematiku. Od formula (1) (2) (3) do valjanih programa za izračunavanje vrednosti hiperboličkih funkcija dug je put. Taj put vodi preko vrednovanja numeričke upotrebljivosti kako tih formula tako i brojnih drugih formula koje sadrže beskonačni proizvod, potencijalni razvoj, racionalni razvoj ili verižni razvoj — i to bez ekonomizacije ili sa ekonomizacijom. Kako nijedna od ovih aproksimacija nije dovoljna za sve vrednosti argumenta neophodno je pribeći kombinacijama nekoliko aproksimacija za svaku računarsku funkciju.

Naravno, uvek je moguće napisati (u smislu matematike) tačne formule (1), (2), (3) — pa šta bude! Ne samo hartija nego i računar trpe sve. Nevolja je jedino što se za većinu mogućih vrednosti argumenta tada ne dobija nijedna tačna cifra rezultata. U korišćenju tako programiranih funkcija „programer" ubrzo uviđa (na svoju bruku) da je pitanje hiperboličkih funkcija za njega postalo hiperbolno. Lakše je naučiti numeričku matematiku nego raditi bez nje.

Hiperbolički sinus

Slika 1 pokazuje da se formula (1) ne može upotrebiti za računanje vrednosti hiperboličkog sinusa sh(x) za male vrednosti modula argumenta. Ako je abs(x) manji od (1/2) log(2) vrednosti exp(x) i exp(−x) su bliske pa je razlika tih vrednosti veoma neprecizna. Relativna greška raste sa smanjenjem abs(x) tako da se za abs(x) $<2^{-J}$ nema nijedna tačna cifra rezultata. Greška predstavljena slikom 1 ne može se smanjiti — jedino je rešenje da se u pomenutoj oblasti argumenta upotrebi neka druga formula. Sa J je označen broj bitova mantise broja. Na primer za „ZX spektrum", „komodora C64" ili Sharp MZ-700 taj broj J je 32. Relativna greška predstavljanja brojeva u računaru je 2^{-J} u pomenutim računarskim brojevi su predstavljeni sa X=0 ili

$$X = \pm 2^k m$$

gde je k karakteristika (ceo broj od −127 do 127) i m mantisa (od 1/2 do $1-2^{-J}$ u koraku po 2^{-J}). Najveći broj koji se može predstaviti je

$$Z = 2^{127}(1-2^{-32}) = 1.701411834E38$$

Za argumente x modula

$$abs(x) \geq log(Z) + log(2) = 128\ log(2)$$

vrednost modula hiperboličkog sinusa je veća od najvećeg broja Z pa treba javiti prekoračenje opsega brojeva (overflow). To se može i posredno učiniti iz programa EXP.

Veliki broj autora predviđa da je u intervalima za

$$log(Z) \leq abs(x) > log(2Z)$$

mogućno izračunati sh(x). Radi se o neopravdanom smanjanju opsega vrednosti funkcija na upola manji interval.

Pojedini autori za velike vrednosti abs(x) koriste aproksimaciju

$$sh(x) = exp(abs(x) - log(2))\ sgn(x)$$

koja sadrži vrlo osetljivu približnu promenu argumenta eksponencijalne funkcije.

Formula (1) ne dozvoljava izračunavanje modula hiperboličkog sinusa u intervalu [Z/2, Z). To su vrednosti koje bi se dobile u intervalima argumenta za koje je

$$127\ log(2) \leq abs(x) < 128\ log(2)$$

Kako se (naravno) želi sprečiti sužavanje oblasti definisanosti računarske funkcije, dovoljno je u tim intervalima uvojiti formulu

$$sh(x) = exp(abs(x) - 88)\ E\ sgn(x),$$

gde je E = exp(88)/2. Preostali član formule (1) je zanemarljiv. On je bio zanemarljiv i ranije. Naime ako je J = 32 onda je exp(− abs(x)) zanemarljivo za

$$abs(x) \geq 16\ log(2)$$

Dakle, u intervalima za koje važi

$$16\ log(2) \leq abs(x) \leq 127\ log(2)$$

može se sh(x) računati sa

$$sh(x) = exp(abs(x))/2\ sgn(x).$$

Za x iz intervala za koje važi

$$(1/2)\ log(2) \leq abs(x) < 16\ log(2)$$

može se hiperbolički sinus računati pomoću

$$s = exp(abs(x))/2,\ sh(x) = s - 0.25/s$$

Za abs(x) < (1/2) log(2) potrebno je naći bolji metod. U matematičkim priručnicima nudi se formula

$$sh(x) = x \prod_{k=1}^{+\infty} (1 + (x/\pi k)^2)$$

Ako se u toj formuli umesto +∞ uzme broj n dobiće se približna formula

$$sh(x) = x \prod_{k=1}^{n} (1 + (x/\pi k)^2)$$

koja ima veoma veliku grešku, videti sliku 2. Kako bi potreban broj množenja bio stotine hiljada i kako želimo da dobijamo rezultat za života programera, ovaj postupak je neupotrebljiv. Neko bi rekao da ova formula ima veliki teorijski značaj, što treba shvatiti ovako: praktičan značaj te formule je nikakav.

Slika 3 pokazuje da je neuporedivo korisniji potencijalni razvoj

$$sh(x) = \sum_{k=1}^{+\infty} x^{2k-1}/(2k-1)!$$

Slika 1. Relativna greška formule
$sh(x) = (exp(x) - exp(-x)) / 2$
ako se funkcija EXP računa tačno.
J je broj binarnih cifara mantise.

Slika 3. Relativna greška formula
$$sh(x) \approx \sum_{k=0}^{n} x^{2k+1} / (2k+1)!\ .$$

Slika 5. Relativna greška razvoja
$$ch(x) \approx \sum_{k=0}^{n} x^{2k} / (2k)!$$

Umesto ovog potencijalnog reda dovoljno je uzeti nekoliko (n) sabiraka. Ako je abs(x) manje od (1/2) log(2) onda je veza između stepena polinoma 2n−1 i broja tačnih bitova B data sledećom tabelom

2n−1	1	3	5	7	9	11	13
B	6	12	21	30	40	50	60

Slika 2. Relativna greška formula
$$sh(x) \approx x \prod_{k=1}^{n} \left(1 + \left(\frac{x}{\pi k}\right)^2\right).$$

Slika 4. Relativna greška formule
$$ch(x) \approx \prod_{k=1}^{n} \left(1 + \left(\frac{x}{\pi(k-1/2)}\right)^2\right).$$

Slika 6. Relativna greška formule
$th(x) = sgn(x)\ (t-1)/(t+1)$
ako se funkcija $t = exp(2\ abs(x))$
računa tačno.

Ekonomizacijom potencijalnog razvoja broj tačnih kitova na istom intervalu argumenta uz isti stepen polinoma (ili druge koeficijente) može se još povećati. Ako se usvoji stepen polinoma (npr. $2n-1=7$) i maksimalna relativna greška (npr. 2^{-32}) onda se ekonomizacijom može proširiti interval primene polinoma na abs(x) manje od 0.38. To je učinjeno u aproksimaciji

Slika 7. Relativna greška razvoja

$$th(x)=\sum_{k=1}^{n} 2^{2k}(2^{2k}-1)B_{2k}x^{2k-1}/(2k)!.$$

$$sh(x)=x+0.1666666667\,x^3+0.0083333\,x^5+0.0001988\,x^7$$

Za male vrednosti modula argumenta dovoljno je usvojiti

$$sh(x)=x.$$

Ako je J=32 (relativna greška $2^{-32}=2.3E-10$) onda se to može učiniti u intervalu

$$abs(x) < (6 \cdot 2^{-32})^{1/2}=0.00004481.$$

Hiperbolički kosinus

Računarsku funkciju hiperbolički kosinus ch(x) lakše je programirati od hiperboličkog sinusa sh(x). Ako je x veće od (J/2) log(2) onda važi ch(x)=sh(x), pa se deo zaključaka o izračunavanju vrednosti sh(x) može koristiti i za ch(x).

Slika 4 predstavlja dokaz da je beskonačni proizvod

$$ch(x)=\prod_{k=1}^{+\infty}(1+(x/\pi(k-1/2))^2)$$

neupotrebljiv u numeričkoj matematici, jer zahteva stotine hiljada operacija množenja. Korisnici programa ne bi bili toliko strpljivi.

Slika 5 pokazuje da se za male vrednosti argumenta potencijalni razvoj

$$ch(x)=\sum_{k=0}^{+\infty} x^{2k}/(2k)!$$

može uspešno primeniti. Granica primene ove formule prema formuli (2) nije vezana za tačnost već za trajanje izračunavanja.

Ako je moduo argumenta mali, onda se za ch(x) može usvojiti

$$ch(x)=1$$

Interval primene te formule je

$$abs(x) < 2^{1-J/2}=0.000021579.$$

Hiperbolički tangens

Računarska funkcija hiperbolički tangens th(x) ne može se računati samo pomoću formule (3)

Slika 8. Relativna greška razvoja

$$th(x)\approx x\Big/\left(\sum_{k=0}^{n} 2^{2k}B_{2k}x^{2k}/(2k)!\right).$$

```
11 LABEL SINH
12 A=ABS(X)
13 IF A< 00004481356354  S=X RETURN
14 IF A< 35 A=A*A S=((A* 0001988+ 008333
333 )*A+ 1666666667 )*A*X+X RETURN
15 ON ERROR GOTO 18  S=EXP(A)* 5*SGN(X)
16 IF A<11 09035489  S=S- 25/S
17 RETURN
18 S=EXP(A-88)*8 258181275E+37*SGN(X)
   RETURN

21 LABEL COSH
22 A=ABS(X)
23 IF A< 00002157518644  C=1 RETURN
24 ON ERROR GOTO 27  C=EXP(A)* 5
25 IF A<11 09035489  C=C+ 25/C
26 RETURN
27 C=EXP(A-88)*8 258181275E+37 RETURN

31 LABEL TANH
32 T=ABS(X)
33 IF T< 00002772785983  T=X RETURN
34 IF T>11 09035489  T=SGN(X) RETURN
35 IF T< 3485735983  T=T*T
   T=X/(1+T/(3+T/(5+T* 1428)))  RETURN
36 T=EXP(T+T)  T=(T-1)/(T+1)*SGN(X)
   RETURN
```

Slika 6 pokazuje da se formula

$$th(x)=sgn(x)\,(t-1)/(t+1),$$

gde je t=exp(2 abs(x)) ne može koristiti za abs(x) manje od (1/2) log(2). Za abs(x) manje od 2^{-J} ne dobije se nijedna tačna cifra rezultata. Ta formula nije upotrebljiva ni za

$$abs(x) > (1/2)\log(2)$$

zbog prekoračenja opsega brojeva (overflow). To je razlog što se računanje th(x) ne svodi samo na sh(x) / ch(x).

Ako je J broj bitova mantise, onda je za

$$abs(x) > (J/2)\log(2)=16\log(2)$$

dovoljna formula

$$th(x)=sgn(x).$$

Za abs(x)<(1/2) $\log_2(2)$ treba naći pogodnu formulu. Za abs(x) < $\pi/2$ važi potencijalni razvoj

$$th(x)=\sum_{k=1}^{+\infty} 2^{2k}(2^{2k}-1)B_{2k}x^{2k-1}/(2k)!,$$

gde su B_{2k} Bernoullijevi brojevi $B_0=1$, $B_1=-1/2$, $B_2=1/6$, $B_4=1/30$, $B_6=-1/42$, $B_8=1/30$, $B_{10}=-5/66$, $B_{2p+1}=0$ za prirodan broj p. Slika 7 pokazuje da je za abs(x) < (1/2) log(2) i J=32 potrebno uzeti prvih 7 članova razvoja.

Slika 9. Relativna greška razvoja

$$th(x)\approx \frac{x}{1+}\,\frac{x^2}{3+}\,\frac{x^2}{5+}\cdots\frac{x^2}{2n-1}.$$

Za abs(x)<π važi razvoj

$$th(x)=x/\left(\sum_{k=1}^{+\infty} 2^{2k}B_{2k}x^{2k}/(2k)!\right).$$

Slika 8 pokazuje da je za abs(x) < (1/2) log(2) i J=32 potrebno uzeti prvih 5 članova razvoja. Broj članova razvoja može biti i 4 ako se izvrši ekonomizacija.

Za proizvoljno realno x važi verižni razvoj

$$th(x)=\frac{x}{1+}\,\frac{x^2}{3+}\,\frac{x^2}{5+}\cdots\frac{x^2}{2n-1}$$

Slika 9 pokazuje da je za abs(x) < (1/2) log(2) i J=32 dovoljno uzeti verižni razlomak reda n=5.

Ako se primeni ekonomizacija, ovi razvoji za isti opseg argumenta i isti broj bitova mantise rezultata J mogu imati i niži stepen (naravno uz izmenu vrednosti koeficijenata). Tako je dobijena formula

$$th(x)=x/(1+t/(3+t/(5+t\ 0.1428))),$$

gde je $t=x^2$.

Ako je J=32 za

$$abs(x) < (3 \cdot 2^{-J})^{1/2}=0.0000277$$

dovoljna je formula

$$th(x)=x.$$

Priloženi su programi za izračunavanje vrednosti hiperboličkih funkcija za računar Sharp MZ-700 (HuBasic). Ako čitalac ima „ZX spektrum" ili „komodora", lako će ubaciti reči LET ili THEN tamo gde su potrebne.

Na osnovu svega izloženog jasno je da je dug put od poznavanja programskog jezika do valjanog programa. Taj put vodi kroz teško prohodni teren koji se zove numerička matematika. Već za tako jednostavne funkcije kao što su hiperboličke funkcije bilo je neophodno kombinovati više formula kako bi se za svaku vrednost argumenta obezbedila tačnost rezultata do poslednjeg bita mantise.

Dušan Slavić

Računari u poslovnoj primeni (3)

zalihe na disketi

Kako postaviti sistem za obradu podataka o zalihama koji omogućuju kako kontrolu nivoa stanja pojedinih artikala na zalihama tako i upravljanje zalihama? Ovako koncipiran zahtev nam omogućuje da opojimo u rešenje tri vrste zaliha sirovine potrošni materijal i rezervne delove. Ove tri vrste zalihe imaju skoro identična sadržaja, tako da ih je moguće rešiti praktično na identičan način. Jasno je pri tome da na personalnom računaru, zbog njegovih ograničenih mogućnosti, ova tri vrste zalihe treba razdvojiti u tri nezavisne obrade.

Kao i u prethodnim slučajevima obradu treba izvesti na računaru koji ima 64 K centralne memorije, serijski štampač i dve disketne jedinice.

ANALIZA ZAHTEVA

Da bi se dao odgovor na postavljena pitanja kada i koliko naručiti, potrebno je izvršiti analizu pojedinih uticaja na same zalihe. Tako na primer pri analizi za odgovor koliko naručiti potrebno je razmotriti nekoliko međusobno protivurečnih činioca

- troškove održavanja zaliha
- troškove naručivanja,
- troškove izazvane kupovinom manjih količina ako se daje niža cena za nabavku kod istog dobavljača (bilo po broju kupljenih komada bilo za iznos ukupne nabavke), i
- troškove zbog gubitaka izazvanih zastarevanjem, kvarenjem ili zbog promene cene

Za analizu ekonomične količine za nabavku na raspolaganju je više metoda, pri čemu one same nisu predmet našeg članka. Međutim, za rad na sistemu je važno da se ta količina unese u sistem kako bi se mogao pojednostaviti proces same nabavke.

S prethodnom analizom je povezano i pitanje kada izvršiti nabavku. Pri tome se mora imati na umu da takozvano vreme ispunjavanja zaliha teče od trenutka kada se da informacija da treba naručiti pa sve do trenutka kada artikal stigne u magacin. Postoji više različitih metoda za određivanje trenutka naručivanja

- unapred utvrđeni nivo ponovnog naručivanja
- utvrđeno vreme pregleda,
- grupne narudžbe,
- prave potrebe,
- rezerve

ili kombinacija dve ili više metoda. U našim uslovima se, uglavnom, primenjuju prve dve metode, pa ćemo ukratko objasniti i parametre koji su kod njih korišćeni.

Pri primeni metode unapred utvrđenog nivoa naručivanja parametri su nivo maksimalnih zaliha i nivo zaliha pri kojoj se vrši naručivanje. Ovaj nivo se određuje na osnovu planirane potrošnje tog artikla za vreme dok traje proces popunjavanja zalihe u skladištu. Pri tome nivo zalihe nikad ne bi trebalo da padne ispod dozvoljenog minimuma (rezervne zaliha).

Kod metode utvrđenog vremena pregleda stanja zalihe parametri su nivo maksimalnih zaliha i utvrđeni vremenski interval kada se kontroliše nivo zalihe (na primer nedeljno ili mesečno). Za razliku od prve metode u kojoj je naručena količina uvek ista (razlika između maksimalne količine i nivoa za ponovno naručivanje) a vremenski interval naručivanja različit, kod ove metode je naručena količina uvek različita, ali se naručivanje vrši u jednakim vremenskim intervalima.

Kada je izabrana metoda kontrole zalihe, i definisane dva parametra koji je karakterišu potrebno je odgovoriti i na pitanje kako voditi cene kod artikla koji je nabavljen sa različitim vrednostima. Moguća su dva rešenja: svođenje cena na poslednju nabavnu cenu ili vođenje višestrukih cena za isti artikal. Teorijski postoji i metoda prosečne cene, ali se ona kod nas praktično ne primenjuje. Od izbora načina iskazivanja cene artikla zavisi i način obrade podataka. Naime u slučaju jedinstvenih cena potrebno je imati program koji će pri svakoj novoj nabavci da izvrši izmenu cene kod artikla koji je nabavljen, te da se odgovarajući kumulativni zbir razlika u ceni usmeri na odgovarajući konto. U slučaju višestrukih cena ovakav program nam nije potreban, ali je zato potrebno šifru artikla na zalihama povećati za jednu cifru na kraju (ako se želi pratiti do 10 različitih cena po artiklu). Na primer osnovna šifra artikla je 05674325, pri čemu šifra 05674325 označava petu cenu a 05674328 osmu cenu za isti artikal. Ovakav izbor rešenja zahteva nešto složenije programe za obradu, ali je u potpunosti ostvarljiv na zadanoj konfiguraciji. Izbor ova metoda, međutim, povlači sa sobom znatno povećanje prostora na disketama. O ovome treba posebno voditi računa u slučaju kada se nabavne cene često menjaju. Rešenje se može naći u periodičnom „čišćenju'' datoteke artikala od slogova sa stanjem nula i odgovarajućim pomeranjem šifara.

Konačno, u zahtevu nije ništa rečeno da li se želi praćenje samo nivoa zalihe ili je potrebno pratiti i analitiku promena po svakom pojedinom artiklu. Ako se želi praćenje samo nivoa zalihe na skladištu, posao je u mnogome olakšan — kako sa stanovišta obrade tako i sa stanovišta potrebnog prostora na disketama.

Datoteka artikala

U našoj obradi postoje dve osnovna datoteka i to datoteka artikala i datoteka dobavljača. U slučaju da je potrebno praćenje i analitika promena po pojedinim artiklima, poželjno je, radi uštede prostora u osnovnoj datoteci artikala, formirati posebnu datoteku naziva artikala. Ovo će sa svoje strane da poveća vreme same obrade, ali će zbog uštede u nazivu artikla (koji je izbačen iz osnovne datoteke) omogućiti praćenje većeg broja promena unutar jednog sloga.

Datoteka artikala bi trebalo da sadrži bar

- šifru artikla (uključujući i jedan znak za varijantu cena i eventualno ako se prati i analitika promena i jedan znak za redni broj sloga)
- naziv artikla
- šifru dobavljača
- maksimalno dozvoljeni nivo zalihe
- minimalno dozvoljenu količinu,
- nivo zaliha pri kome se vrši naručivanje odnosno datum (na primer svaki 10-ti dan ili 30-dan u mesecu) zavisno od izbora metode za naručivanje
- trenutni nivo zalihe
- jediničnu cenu,
- jedinicu mere (šifrirano),
- naručenu količinu,
- datum poslednjeg izlaza artikla iz skladišta i
- rezervisanu količinu

Ovim pregledom bi se obuhvatio minimalan broj podataka u slogu za zalihe ako nam nije potrebna analitika promena. Ukoliko se pak želi i analitika promena poželjno je formirati novu datoteku koja bi sadržala šifru artikla, naziv artikla i šifru dobavljača kako bi se u slogu oslobodio prostor za podatke o analitici. Takođe je moguće i formirati novu datoteku koja bi sadržala samo promene, a osnovna datoteka ostala kao što je već navedeno. Izbor između ova dva mogućnosti je, u suštini, izbor između brzine vremena obrade i raspoloživog prostora na disketama. Verovatno, u većini slučajeva ćemo žrtvovati vreme obrade, jer smo ipak sa prostorom na disketi jako ograničeni.

Analitika promena

Podaci koje bi bilo poželjno pratiti u analitici promena su

- datum promene
- iznos promene i
- šifru promene

Kako se ovi podaci javljaju u jednakim razmacima unutar sloga, to je potrebno uvesti i indikator koji ukazuje na prvi slobodan element za promenu koja se očekuje.

Ako uz prva tri podatka za svaku promenu dodamo i podatak

- mesto troška (ili broj radnog naloga po kome je roba izdata)

Dva do sada prikazana primera primene personalnih računara u radnoj organizaciji mogu čitaoca navesti na pomisao da se ovakvi računari primenjuju samo na nekim „sporednim" poslovima, koji nisu vezani neposredno za proces proizvodnje. U primeru koji opisujemo u ovom broju, kontrola i upravljanje zalihama, pokušaćemo da prikažemo mogućnost primene u oblasti koja ima veoma veliki uticaj na sam proces proizvodnje. Pri tome se nećemo obazirati na izvesne razlike, koje svakako postoje, uslovljene različitim vrstama zaliha — sirovina, polufabrikati, gotovi proizvodi, ambalaža, potrošni materijal, rezervni delovi, itd. — već ćemo se zadržati samo na opštem opisu kontrole i upravljanja zalihama, a zainteresovani čitalac će svakako moći da, uz malo imaginacije, postavi i svoj problem u okvir prikazanog rešenja.

moguće je pratiti i odgovarajuće troškove na željenim mestima u proizvodnji.

Dodajući još i šifru artikla, dobićemo i konačan sadržaj datoteke promene.

Jasno je da ove datoteke moraju biti organizovane kao indeksne datoteke, kako bi nam bilo omogućeno da pristupimo svakom artiklu po želji. Osim toga, ovako organizovane datoteke nam omogućuju da bez većih problema ulančavamo manje slogove u jedinstvene celine.

Osnovni problem koji se ovde javlja je veličina samih datoteka, odnosno ograničen kapacitet koji imamo na disketama. U slučajevima kada radimo sa oko 2000 artikala, verovatno da ne bi trebalo očekivati nedostatak prostora na jedinicama sa direktnim pristupom, ali ukoliko imamo veći broj artikala i, uz to, želimo da radimo i analitiku promena, verovatno da ćemo datoteku artikala morati da proširimo na više disketa. Ovo, sa svoje strane, uglavnom utiče na komfornost rada, ali ne i na mogućnosti koje su nam na raspolaganju. Pri tome, svakako, ne mislimo na količine koje se mere desetinama hiljada artikala. Ukoliko to i jeste slučaj, moguće je artikle podeliti ili po računopolagačima ili po vrstama robe (na primer, metalna, elektro proizvodi, od gume i slično), pa ih tako razvrstane postaviti na različite diskete ili pak na više malih računara. Ulaganja uglavnom nisu velika, ali su koristi u odnosu na ručni rad nemerljive.

Upravljanje zalihama

Na ovaj način formirani sadržaji datoteka nam omogućuju dobijanje većeg broja izveštaja koji se mogu koristiti u procesu kontrole i upravljanja zalihama. Bez ulaženja u opise pojedinih rezultata obrade, navešćemo samo neke izveštaje koji se mogu dobiti:

- ABC klasifikacija robe na zalihama,
- popisne liste,
- izveštaj o artiklima koji stoje određeno vreme u magacinu a nisu trebovani,
- izveštaji o zalihama koje su ispod minimuma,
- pregled vrednosti zaliha,
- izveštaji o potražnji pojedinih artikala,
- izveštaj o koeficijentima obrta pojedinih artikala,
- izdavanje naloga za nabavku,
- praćenje i prijem naručene robe,
- praćenje troškova po mestu troška ili radnom nalogu,
- dobijanje analitičke kartice promena za svaki artikal,
- materijalni bilans za buduću proizvodnju (u posebnom procesu obrade).



— Bili su u pravu! Kompjuteri zaista olakšavaju vođenje inventara!

Većina ovih izveštaja se dobije sekvencijalnim čitanjem datoteke artikala, tako da je procena vremena obrade za izradu jednog izveštaja teško ostvarljiva bez prethodnog poznavanja veličine same datoteke. Međutim, uzimajući u obzir da je štampač usko grlo našeg sistema, realno je očekivati da bi vreme za izradu pojedinih izveštaja, uključujući i vreme potrebno za manipulisanje sa disketama, bilo reda veličine desetak minuta. Pri tome se izuzimaju izveštaji kao što su popisne liste ili, pak, izveštaj o koeficijentima obrta za sve artikle u skladištu. Poseban problem je vreme potrebno za izlistavanje kompletne analitike promena svakog artikla. Ovo vreme može biti tako dugo da realno, za jako velike datoteke i veliki broj promena, može biti neprihvatljivo. U svakom slučaju, moguće je računati da je vreme potrebno da se odštampa jedan red na prosečnom štampaču, oko 1 sekunda. S druge strane, izvestan broj izveštaja ili informacija se može dobiti praktično trenutno (na primer, stanje zaliha određenog artikla) ili u toku same obrade ulaznih podataka (prijem naručene robe).

Personalni računari mogu, očigledno, imati svoje mesto i u poslovima vezanim za kontrolu i upravljanje zalihama. Pri tome treba biti obazriv pri postavljanju zahteva ukoliko se radi o velikom broju artikala koje želimo da pratimo i analitički. U tom slučaju je verovatno najpogodnije izvršiti prethodno ABC klasifikaciju, a zatim na sistem postaviti ili samo klasu A ili eventualno i klasu A i B, dok bi klasa C ostala da se i dalje vodi ručno, odnosno na način kao i do sada.

Mihajlo Karapandžić, dipl. ing.

U SVETU KOMPONENTI

Šta ima novo

Era CMOS-a

U skladu sa predviđenjima eksperata komplamentarna MOS (CMOS) tehnologija sva više postaje vodeća tehnologije u elektronici 80-tih godina sa trendom da to ostane i sledeća dekade Praktično nama poznatog tipe mikroprocesore ili vrsta memorija koji se ne proizvode kako u izvornoj NMOS tako i u CMOS verziji

Z80 u CMOS-u

Zanimljivost za korisnika popularnog Z80 mikroprocesora japanski proizvođači (inače vodeći u CMOS tehnologiji) a pre svih firme TOSHIBA i SHARP, proizvode CMOS verzija Z80 CPU, kao i perifernih kola iz familije (PIO CTC, SIO a uskoro i DMAC) dok TOSHIBA ima i specijalni CLOCK generator CMOS Z80 (TOSHIBA oznaka TMPZ84COOP) izrađen C²MOS Silicon Gate tehnologijom je potpuno hardverski (pin-for-pin) i softvarski kompatibilan sa izvornim ZILOG Z80 CPU, ali ima prednosti koje inherentno sadrži CMOS potpuno statički rad i ekstremno niska potrošnje (15 mA na 5 MHz i +5V aktivno odnosno samo nekoliko mikroampera u HALT-u, idealno za portabl uređaje!)

Kompletirana familija

INTEL je kompletirao svoju seriju 16-bitnih mikroprocesora tipom IAPX 286/10 (oznaka 80286) koji, kao i njegovi prethodnici IAPX 86/10 (8086) i IAPX 186 (80186) ima svoj numerički koprocesor (80287) poboljšani kontroler za direktan pristup memoriji (82258 ADMA) LAN (Local Area Network) koprocesor (82586/8) itd 80286 je mašina sa virtuelnom memorijom (integrisani memory Management Unit — MMU adresira do 1 GByta) pristup fizičkoj memoriji je do 16 MByta radi sa CLOCK-om od 6 MHz izvršava superset 8086 instrukcija i pogodan je za egzekuciju operativnih sistema ili viših programskih jezika Takođe ga proizvode (tzv Second Source) i AMD i SIEMENS tako da zajednički utiru put punom 32-bit CPU koji se uskoro očekuje (80386)

Tridesetdvobitne laste

MOTOROLA je jedna od prvih firmi koja je iznela na tržište puni 32-bit CPU (MC 68020) iz već čuvene 68000 familije koji uz standardni 16-bit 68000, zatim MC 68010 (virtuelna mašina) i MC 68008 (8-bit DATA-BUS) čine snažan tim za svaku aplikaciju U tome mu pomažu periferna kola kao MC 68120/1 IPC MC68230 Timer, MC68450 DMAC, MC68451 MMU MC68488 GPIB adapter itd , a sada i MC68681 Dual UART i MC68881 Floating Point Coprocessor (FPU koji odgovara IEEE P754 predlogu) MOTOROLA 680XX seriju po licenci proizvode i SIGNETICS/PHILIPS, MOSTEK THOMSON, HITACHI i ROCKWELL Pa ipak, najviše periferija za 68000 ne proizvodi MOTOROLA koja je originalno napravila 68K CPU (a zajedno sa MOSTEK-om i SIGNETICS-om osmislila tzv VMEbus) već SIGNETICS koji na programu ima petnaestak LSI kola od 68154 IGOR do 68920 MAC

Procesori na bejziku

Onima koji mikroprocesore koriste za kontrolerske aplikacije, a ne žele da se upuštaju u mašinsko programiranje, već izvesno vreme stoje na raspolaganju CPU sa inkorporiranim bejzik interpreterom na čipu (uprošćeni tzv TINY BASIC) ZILOG ima Z8671 (2K ROM preprogramirana verzije Z8 CPU) a NATIONAL SEMICONDUCTOR INS8073 (4K ROM preprog INS8070 CPU) dok NEC pravi uPD7901 Ovi „bejzik interpretari na čipu ' imaju standardne bejzik instrukcije, ali rade sa INTEGER brojevima Imaju sada konkurenciju u INTEL-u (preprogramirana verzija 8052 CPU sa 8K ROM-a ima Floating Point BASIC)

Brza CMOS kola

Dugo smo koristili standardna TTL kola (74 seriju), brza TTL (Schottky, 74S) i brza sa malom potrošnjom (Low Power Schottky, 74LS), danas se sve više upotrebljavaju poboljšana brza TTL kola (74AS, sa 40% većom brzinom rada) ili 74ALS serija sa 80% manjom potrošnjom Međutim, izgleda da će ih u nekoliko sledećih godina potpuno zameniti brza CMOS kola (74HC sa CMOS logičkim nivoima, odn. 74HCT sa TTL nivoima) zbog svojih dobrih karakteristika brzine rada koja je otprilike jednaka brzini TTL kola (ranije je to bila glavna mana CMOS tehnologija), cena koštanja (jedva nešto skuplja od ekvivalentnih TTL kola), ekstremno mala potrošnje (za nekoliko redova veličine), znatno veća pouzdanosti (posledica niska potrošnja statističko vreme između otkaza reda stotina, pa i hiljada godina!), širokog opsega napona napajanja, velike ulazna impedanse simetričnog respona izlaznog napona (jednak radnom naponu), itd

Brzi .

ZiLOG je znatno snizio cenu brze verzije Z80 CPU (oznaka Z80H) koje radi sa CLOCK-om od 8 MHz, sa nadom da će ovo doprineti boljoj prodaji i široj upotrebi (protivnik je bila visoka cena)

. . . i modifikovani Z80

AMI proizvodi S83 OSP (Operating System Processor) koji se u suštini sastoji od Z80 CPU plus kontroler za dinamička RAM-ova plus 8K ROM (masked) koji na smeta spoljnoj memoriji (adresibilnost i dalje 64K bajta), interni ROM služi za smeštanja rutine sopstvenog REAL-TIME operativnog sistema

Eprom sa lozinkom

Iako se današnji ultravioletni EPROMi prave sa kapacitetom od pola megabita (organizacija 64K×8, i ovde je naravno, prvi bio INTEL) sa planom za megabit i više, posle invazija CMOS verzije (naročito japanskih firmi, sa maksimalnim kapacitetom od četvrt megabita kod 27C256), INTEL je pripremio i nešto drugačiji odgovor na tržištu se pojavio prvi 128K KEPROM (oznaka 27916, KEY EPROM) kojem se može pristupiti samo ako poznajete šifru zapisanu u njemu Ovaj tip memorija sadrži pored EPROM-a, LOCK/KEY deo, generator slučajnih brojeva i deo za šifrovanje

68000 na japanski način

Firma HITACHI je smestila popularni HD68000 CPU u znatno manje i ekonomičnije kućište tipe PLCC sa 68 pinova (Plastic Leadless Chip Carrier, novi JEDEC standard)

Od jednog do dvesto

Raspon cena modernih mikroprocesora pokazuje zavidnu dinamiku dok neke verzija mikrokontrolera koštaju ispod jednog dolara (npr 68XX plastična verzija OEM količina), INTEL-ov 80286 košta oko 200 US dolara (naravno vojne verzije mikroprocesora visokih performansi — 16/32 bit — prema MIL-STD-883 sa dodatnim testovima koštaju i znatno više)

Mostak pred bankrotstvom

Sveža vest firma MOSTEK (poznata po dinamičkim memorijama i mikroprocesorima) je bankrotirala ali se francuski gigant THOMSON—CSF sprema da kupi

Pripremio: Blažimir P. Miše, dipl ing.



Profi adresar

U Računarima 10 pisali smo o društvima programera koja okupljaju profesionalce. Od njih se, razume se, ne mogu dobiti katalozi najnovijih igara za spektrum niti cenovnik novih dodataka za komodor. Oni odista svakom programeru koji drži do sebe pružaju mnogo dragocenije — informacije i priliku da unapredi svoje tehnike programiranja. Možete tako da posećujete razne konferencije, simpozijume i izložbe koje to društvo organizuje po izuzetno povoljnim uslovima, ili da se pretplatite na računarske časopise po izuzetno povoljnim cenama, da kupujete knjige sa popustima i od 70%, da dobijete stipendiju od društva za usavršavanje ili da položite određene ispite koji se visoko cene pri zapošljavanju. Za ostale pogodnosti raspitajte se kod samog društva.

Ako se odlučite da nekom od navedenih društava pišete, možete da prepravite pismo u kome drug Sofronije piše Association for Computing Machinery. Naveli smo samo obavezni deo teksta, a na mesto tri zvezdice (***) možete da napišete nešto što će doprineti društvu da stekne bolji uvid u vaše želje i kvalifikacije. Društva su vrlo ekspeditivna i odgovaraju na sva pisma i hakera i akademca.

American Federation od Information
Processing Societies Inc (AFIPS)
Suite 800
1815 N Lynn Street
Arlington VA 22209
USA
(702) 558-3600

Association of Computer Users (ACU)
1690 38th Street
P O Box 9003
Boulder CO 80301
USA

Association for Computing Machinery (ACM)
11 W 42nd Street
New York NY 10036
USA
(212) 869-7440

Association for Educational Data Systems (AEDS)
1201 16th Street NW
Washington, DC 20036
USA
(202) 833-4100

American Institute of Aeronautics and Astronautics (AIAA)
1290 Avenue of the Americas
New York NY 10104
USA
(212) 581-4300

American Mathematical Society (AMS)
P O Box 6248
Providence RI 02940
USA
(401) 272-9500

American Society for Information
Science (ASIS)
1010 Sixteenth Street NW
Washington DC 20036
USA
(202) 659-3644

American Statistical Association (ASA)
806 15th Street NW
Washington DC 20005
USA
(202) 393-3253

Association for Systems Management (ASM)
24587 Bagley Road
Cleveland OH 44138
USA
(216) 243-6900

Association of Computer Programmers
and Analysts (ACPA)
c/o Cute Corp
Suite 808
11800 Sunrise Valley Drive
Reston VA 22091
USA

Association of Data Processing Service
Organization (ADAPSO)
Suite 300
1300 N 17th Street
Arlington VA 22209
USA
(703) 522-5055

CAUSE The Progessional Association for the
Development Use and Management of
Information Systems in Higher Education
737 29th Street
Boulder CO 80303
USA
(303) 449-4430

Computer and Automated Systems Association
Society of Manufacturing Engineers
(CASA/SME)
One SME Drive
P O Box 930
Dearborn, MI 48128
USA

Computer and Business Equipment
Manufacturers
Association (CBEMA)
1828 L Street NW
Washington DC 20036
(202) 466-2288

Computer and Communications Industry
Association (CCIA)
Suite 512
15500 Wilson Boulevard
Arlington VA 22209
USA
(703) 524-1360

Independent Computer Consultants Association
(ICCA)
P O Box 27412
St Louis, MO 63141
USA
(314) 567-9708

Institute for Certification of Computer
Professionals (ICCP)
35 East Wacker Drive
Chicago IL 60601
USA
(312) 782-9437

Institute of Electrical and Electronic Engineers
(IEEE) Computer Society
Suite 201
1109 Spring Street
Silver Spring MD 20910
USA
(301) 589-3386

Institute of Internal Auditors (IIA)
249 Maitland Avenue
Altmount Springs FL 32701
USA
(305) 830-7600

Instrument Society of America (ISA)
P O Box 12277
78 Alexander Drive
Researc Triangle Park NC 27709
USA
(919) 549-8411

Association for Computing Machinery
11 W 42nd Street
New York NY 10036
USA

Sofronije Hadži-Janićijević
Medvedgradska 20
21000 Novi Sad
Yugoslavia
January 15 1986

Dear Sir

*My name is Sofronije Hadži-Janićijević *** I am very interested in joining the (ime društva). Please send me all information needed to become a member at the above address by air mail*

Sincerely
(potpis)
Sofronije Hadži-Janićijević

Computer Law Association Incd (CLA)
8106 Lorcom Court
Springfield VA 22182
USA
(202) 272-2906

Computer Security Institute (CSI)
43 Boston Post Road
Northboro MA 01532
USA
(617) 393-3663

Conference Board of the Mathematical
Sciences (CBMS)
1500 Massachusetts Avenue NW
Washington DC 20005
USA
(202) 293-1170

Data Processing Management Association
(DPMA)
505 Busse Highway
Park Ridge IL 60068
USA
(312) 825-8124

EDUCOM The Interuniversity Communications
Council Inc
P O Box 364
Princeton NJ 08540
USA
(609) 734-1915

International Federation for Information
Processing (IFIP)
IFIP Secretariat
3 Rue du Marche
CH-1204 Geneva
Switzerland

International Information/Word Processing
Association (IIWPA)
1015 North York Road
Willow Grove PA 19090
USA
(215) 657-6300

Mathematical Association of America (MAA)
1529 Eighteenth Street NW
Washington DC 20036
USA
(202) 387-5200

National Computer Graphics Association, Inc
(NCGA)
Suite 601
8401 Arlington Blvd
Fairfax VA 22031
USA

National Science Foundation (NSF)
Division of Mathematical and Computer
Sciences
Room 339
1800 G Street NW
Washington, DC 20550
USA
(202) 632-5960

N. Aleksić

mali oglasi

Ako ne možete da podnesete da drugi nemaju ono što vi nemate javite se na neki od malih oglasa u „Računarima".

Ako ne volite da se dopisujete sa „Računarima" svoj mali oglas možete nam izdiktirati preko telefona 011/650-161 svakog radnog dana od 10—14 sati

Prva stvar koju treba da uradite je da se odlučite da li želite običan ili uokviren mali oglas.

CENA OBIČNOG MALOG OGLASA do dvadeset reči je 600 dinara. Svaka naredna reč košta 40 dinara s tim što oglas ne sme da ima više od 50 reči. Adresa oglašivača ne računa se u cenu.

CENA UOKVIRENOG MALOG OGLASA je 600 dinara po visinskom centimetru s tim što se mogu zakupiti najmanje po 32 slovna znaka. Ako se ne iskoristi čitav prostor u jednom redu računa se broj redova a ne broj znakova. Za uokvirene oglase preko 5 cm cena je 900 dinara po centimetru.

Poželjno je da vaš mali oglas počinje sa Prodajem, Kupujem, Držim časove, Menjam ... ili načim sličnim što ukazuje na sadržaj oglasa.

Svi mali oglasi koji stignu u našu redakciju do 30 januara do 12 sati biće objavljeni u „Računarima" broj 12 koji izlaze iz štampe polovinom februara.

Da ne bi bilo zabune obavezno naznačite da li želite običan ili uokviren mali oglas i zajedno sa tekstom vašeg malog oglasa pošaljite i priznanicu o uplati na adresu redakcije „GALAKSIJA", BULEVAR VOJVODE MIŠIĆA 17 „za male oglase".

Vaš mali oglas možete nam izdiktirati preko telefona 011/650-161, a mi ćemo vam onda naknadno poslati ispunjenu uplatnicu.

„RAČUNARI" OBJAVLJUJU NAJJEFTINIJE RAČUNARSKE MALE OGLASE!!!

OBRADUJTE SVOJ RAČUNAR! OBJAVITE MU MALI OGLAS U „RAČUNARIMA"!

SPEKTRUM

• SPEKTRUM — Rečnik englesko srpskohrvatski (oko 1400 najčešće upotrebljavanih reči) + kaseta + poštarina (1300 din.) Tel. (011) 497-662 od 17-19 h. Dragan Marjanović 11000 Beograd B. Jerković 123/36

• Spektrumovci! Proffessional software vam nudi sve što vam je potrebno. Posedujemo najbolja strane i sva nova programa (Impossible Mission, Hackar, WS Baskatball ...). Tražite besplatan katalog. Uverite se da su nam cene među najjeftinijim u Jugoslaviji. Novi programi stalno pristižu. Proffessional software Nušićeva 1 15000 Šabac tel. 015/24-734

• P N P electronic HAROVER ZA SPEKTRUM POVOLJNO PRODAJEMO KEMPSTON JEDNOSTRUKE I DVOSTRUKE INTERFEJSE ZA PALICU PALICE I/O INTERFEJS A/D D/A KONVERTORE SVJETLOSNU OLOVKU EPROM PROGRAMATOR, MEGAROM AUDIO POJAČALO PREPRAVLJENI ROM ZA SPEKTRUM PROGRAMIRAMO EPROME POPRAVLJAMO SPEKTRUM I KOMODOR JOŠ MNOGO TOGA BESPLATAN KATLOG PNP electronic Jerotova 12 Split

• Za Sepktrum prodajem programe za radio-amatere ERTTY SSTV (rad bez interfejsa) ORB OTH CV KEY BOARD i mnogo toga korisno radio-amaterima. Prodajem najnovija programe za igru (decembarski hitovi u Engleskoj). Branković Rade P. Box 37 12240 Kučevo

• SPEKTRUM — OBAVEZAN PRIRUČNIK ZA POČETNIKE I NAPREDNE BASIC PROGRAMIRANJE i brošura UVOD. Do sada nenadmašna kvalitata! DUŠKO BJELOTOMIĆ CENTAR 1 54550 VALPOVO tel. 054/82-665 ili 041/683-141
3 cm

• Prodajem programe za Spektrum. Komplet od 14 programa 700 din + vaša kaseta. Sa mojom kasetom 1000 din. Milan Vunjak Mrakovička 50/8 11000 Beograd tel. 535-532

• Najbolja ponuda za Spektrum povoljna cena novih programa najniža cena skripti i uputstava. Popust. besplatan spisak. Sinadinović Dragan Gundulićeva 12 34300 Aranđelovac tel. 034/714-948
3 cm

• Besmrtni poka katalog za Spektrum. Preko 100 pokova za igre — 200 d. Sada možete igrati do kraja. Toi Tibor B. Dajanović 1 21205 Sremski Karlovci tel. 021/881-650

• SPECTRUM — SERVIS Servisiranje svih kvarova na Spektrumu za samo 3300 d. Cena je fiksna a u nju su uračunati i delovi. Spektrum možete poslati poštom. Rok sevisiranja je 24 časa. Bećarovski Aco Gr. zid k. 12/40 91000 Skopje tel. 091/239-551

• Super Turbo II ubrzava učitavanja — snimanja. omogućava presnimavanje programa zaštićenih ubrzanim učitavanjem (Program, kaseta uputstvo — 1000 d.) Komplati najnovijih programa 800 d. Tražite katalog. Kralić Aleksandar Balzakova 30 21000 Novi Sad

• SPEKTRUM UBEDLJIVO NAJJEFTINIJI PAKETI NA YU TRŽIŠTU — ZAHTEVAJTE NOV BESPLATAN KATALOG SA PREKO 1000 PROGRAMA. Radulović Rade Vožatski put 10 61000 Ljubljana tel. 061/225-568

• SPEKTRUM Profesionalni prevodi MAŠINAC ZA POČETNIKE (1000) DISASEMBLIRANI ROM (1200) NAPREDNI MAŠINAC (1300). KOMPLET (3100) Prevedena uputstva za prevedene programe DEVPAC, MEGA BASIC BETA BASIC 1.8 ARTIST MELBOURNE DRAW QUILL MONITOR DISASEMBLER EDITOR ASEMBLER FIFT LEONARDO. Pojedinačna uputstva (500) animiljani programi (200) Bilo kojih pet uputstava + animiljani programi (2500) Isporuka za 24 časa. KOMPJUTER BIBLIOTEKA Filipa Filipovića 41 32000 ČAČAK tel. 032/31-20

VRHUNSKI GRAĐEVINSKI PROGRAMI za ZX Spektrum OKVIRI REŠETKE ROŠTILJI PILOTI DIMENZIONIRANJE VODOVOD i mnogi drugi. Za radne organizacije i pojedince Besplatan katalog
Gino Gracin Kosala 17
51000 RIJEKA
tel. 051/517-291

• Svakog mjeseca najnoviji ZX Spektrum programi u polusatnim kompletima (po šest programa) za samo 160 dinara. Tražite katalog sa 800 programa. Marko Marković Dž. Bijedića 27A/XI 71000 Sarajevo tel. 525-212

• Prodajem programe za Spektrum. Jedan (bilo koji!) program košta 50 dinara. Na svakih pet kupljenih programa jedan dajem kao poklon. Pišite za besplatan katalog! Janković Šilard Petefi Šandora 84 25222 Telečka

• KEMPSTON ITERFACE (KEMPY)
REDOSTIK PALICA (REDY)
CIJENA KOMPLETA —— — 12 000 din

SPEKTRUMOVCI! I u Novoj godini — najnoviji i najkvalitetniji kompleti sa dvanaest programa po 1000 dinara plus kaseta uz velike popuste za veća narudžbe. Komplet 21 Popeye W.S. Basketball Superpipelina 2 Komplet 22 Winter sports Commando Robin of the Wood komplet 23 Saboteur Yie Ar Kung fu Imposible Mission. Tražite katalog s opisima (50 d.) sa naznakom za Spektrum. Snimka profesionalna isporuka ekspresna! FUTURE ORION 41000 ZAGREB Rubalićeva 7 tel. 041/417-052 POSEBAN PROGRAM ZA ODRASLE Sex Mission Zajedno sa kasetom i troškovima otplata 950 d.

• SPEKTRUMOVCI novi hit-programi (48K) Komplet 7 Skool Daze 2 Beach Head 2 Boulder Dash 2 International karate 1 International karate 2 Impossible Mission Sorderons Shadow Dynamite Dan Monty on the Run Exploading Fist. Dundarach Frankie. Komplet + kaseta + poštarina, 1400 d. Garantovan kvalitet i brza isporuka. Narudžbina na adresu. Nastasović Srđan Oslobođenje II deo 6 11194 Beograd — Rušanj ili na tel. 011/858-222

NEW DATA

• SPECTRUM — AMSTRAD — SCHNEIDER
Programski paketi za stručnjake
STATISET
LINSET
GEODET
NEW DATA D. Brašovana 8/10
21000 Novi Sad
SCHNEIDER — AMSTRAD — SPECTRUM

SPECTRUM RAINBOW SOFTWARE vam nudi izbor od preko 1500 programa. Kompleti od 25 programa 800 d. Sve najnovija programa koja drugi nude kod nas možete dobiti za samo 100 d. Svake nedelje novi komplet. Rainbow software snima direktno iz računara. Katalog je besplatan. Mihajlovski Krćo Moše Pijade 128 91300 Kumanovo tel. 091/23-800

• ZPS SOFTWARE — još prošle godine dobili smo ove hitove za Spektrum: COMMANDO ROLLER COASTER SABOTER NEVER ENDING STORY BACK TO SKOOL BEACH HEAD 2 WINTER SPORTS ROBIN ON THE WOOD YIE AR KUNG FU i još mnogo drugih. Informacije o najnovijim programima na tel. 041/670-071 Borislav Vahčić Viktora Kabarića 36 41020 Zagreb

EPROM PROGRAMATOR MEGAROM LAJT PEN, PRINTER INTERFEJSI I JOŠ MNOGO TOGA
P N P electronic Jerotova 12 Split

MASTERCLUB
FOR SPECTRUM & CBM 64 COMPUTERS

• Spectrum news
1 COMMANCO (Elite) arkadni rat
2 ROLLER COASTER (Elite), luna park
3 YIE AR KUNG FU (Imagine), kung fu II
4 RASPUTIN (Firebird) bolji od knight lora
5 S C OUEST (Software project) konačno na Spektrumu
Tu su naravno i BEACH HEAD II, BOULDER DASH II, SABOUTHER, INTERNATIONAL KARATE
Kao što vidite MasterClub je izabrao za Vas samo najbolje. Ako vas zanimaju apsolutni noviteti pogledajte što smo pripremili za Vas u januarskom broju MOG MIKRA. Novosti za Vaš SPECTRUM I COMMODORE 64 Obavezno 50 nd za katalog.
MASTERCLUB ZAGREBAČKA 21, RIJEKA 051/37-546

SPECTRUM SOFTWARE STUDIO
— Izbor od preko 1000 programa
— svaki program sa uputstvom
— veliki izbor literature — knjiga i originalnih programskih uputstava na engleskom i srpskohrvatskom jeziku
— spisak programa je besplatan, za katalog sa opisom poslati 200 d
— brza i kvalitetna usluga — proverite!
PAJNIĆ MIRKO
STRAHINIĆA BANA 56 11000 BEOGRAD
Tel 011/188-190 posle 15 časova

• Spektrumovci! Najmoćnije logaritamske tablice od sada i za vaš kompjuter Kaseta sa programom + kompletna uputstva za samo 700 din Logos Soft — Vujčić Zoran, Retke Vujovića 2/XII 11090 Beograd tel 011/532-308

• Spektrumovci! Povoljno prodajem najnovije programe pojedinačno i u kompletu Najnoviji super komplet sa 12 hit programa INTERNACIONAL KARATE NEVER ENDING STORY WRIGGLER SUPER BRAT (Match Point 2) Nerlović Srđan Braće Santini 54 58000 Split tel 058/512-640

• Spektrumovci! Mega prilike! Sve one igre za koje kažu da su najbolje u 1985 (Računari 10) osim Gyrona D Bueters DT Dechatlon ali sa SEX MISSION IMPOSSIBLE MISSION FIGHTING WARRIOR i INTERNATIONAL BASKETBALL prodajem za samo 800 din + kazeta Marko Premzl Rade Vene 5, 43260 Križevci tel 043/841-792

• Spektrumovci! Nudimo širok izbor najnovijih superhitova u polučasovnim kompletima po izboru WINTER GAMES INTERNATIONAL BASKETBALL SABOTER čekaju na vas Tražite besplatne kataloge Tel 345-844

COMET SOFTWARE
Sva za ZX SPEKTRUM
— video igre
— namenski programi
— programska uputstva
— literatura
— SPECIJALNO —
BETA BASIC 3 0 sa kompletnim originalnim uputstvom kasetom i poštarinom za samo 1000 -d

Za sve usluge garancija!
MILOVANOVIĆ LJUBIŠA
Petra Lekovića 57 11030 BEOGRAD
tel 011/558-007 posle 17 časova

• CHALLENGER SOFTWARE — za Spektrum prodajem IMPOSSIBLE MISSION (program godine) BEACH HEAD 2 (US Gold) BOULDER DASH 2 (nezaustavljivo padanje stena), MACADAM BUMPER (fliper po meri) SKOOL DAZE 2 (izvrsno), WARIOR (exploding fist 2)+još šest super hitova Samo 600 din Čeda Nedeljković XI krajiške divizije 61 11000 Beograd, tel 011/538-612

• EMP SOFTWARE vam predstavlja najnovija igre engleskih top lista za Spektrum (izvrsni BASKETBALL RED ARROWS) Za svaki komplet programa dobijate objašnjenje u obliku kataloga (besplatno)+besplatan katalog Koser Igor Augusta Cesarca 4 54400 Čakovo tel 054/841-812

• Izaberite si najnovija programe za ZX Spektrum Za više od 20 programa dobijete popust od 10% Programi se prodaju pojedinačno i u kompletima Tražite besplatan katalog Tel 041/686-747 i 041/672-061

• SPEKTRUMOVCI! Nervira vas sporo učitavanje TURBO TAPE za SPEKTRUM u dvije verzije Prva ubrzava učitavanje i snimanje na 3000 bauda i jednostavna je za rukovanje dok druga verzija radi na 3600 i 7200 baude Cijena pojedinačne verzije je 1000 din Ako kupite obe cijene tada iznosi 1700 din Tel 041/686-747

• Ako još ne posedujete generator zvuka za vaš ZX Spektrum ovo je idealna prilika da ga nabavite! Cena 15000 dinara Dražen Abramović, Bjelovarskog odreda 9, 41020 Zagreb

• Spektrumovci komplet 17 — komplet snova Popeye W S Basketball Monty on the run Tiranenog 3 Fehrenheit, S Pipeline 2 Dynemite Dan A S Profanation Red Arrows Drive in T T Spirit Day in life Confusion Sva to za 700 din + kaseta Posedujem i mnoštvo starijih kompleta! Perić Nenad Braće Miladinov 12 37000 Kruševac tel 037/33-510

• Spektrumovog najvećeg neprijatelja — pregrijevanje možete savladati ako naručite precizne upute za samogradnju stabilizatora koji će produžiti život vašem ljubimcu Cena prava sitnica Branko Vukasović Reva Jenković 8 71000 Sarajevo

• Svakog meseca najnoviji ZX Spektrum programi u polusatnim kompletima (po šest programa) za samo 180 dinara Tražite katalog sa 700 programa Marko Marković Dž Bijedića 27A/XI 71000 Sarajevo, tel 525-212

• Prodajem ZX SPECTRUM 16 K sa kompletnim priborom i priručnikom na njemačkom Cijena 50 000 00 dinara Zoran Kahrić Virovitica tel 046/721-523

• Spectrum — multisoft uvek sa vama Exploding fist D T Superleel Pentagram itd Dragan Jovanović Dragoslava Jovanovića 12, 37000 Kruševac tel 037/30-568

• Spectrum Rainbow Software vam nudi super najnovija programe po najpovoljnijoj ceni Izbor od preko 1500 programa Kompleti od 25 programa 600 din Katalog besplatan Mihajlovski Kirčo Moše Pijade 129 91300 Kumenovo tel 0901/23-900

• NAJJEFTINIJE!!! BEZ KONKURENCIJE!!!
SPEKTRUM — Komplet od 170 programa — 1 700 din na vašim ili 3 000 din na moje 4 kasete Sevinovski Saša Gajeva 4 43400 Virovitica

• SPEKTRUM — Najbolji i najnoviji programi u kompletima od 14—20 programa — 600 d na vašoj ili 1000 d na mojoj kaseti Sevinovski Saša Gajeva 4 43400 Virovitica

• Sretnu Novu godinu želi vam El Toro Software! Uđite u Novu godinu sa najboljim programima za ZX Spektrum Winter Games Yie ara kung fu Back to skool Beta basic 3 0 International karate Pišite za katalog sa slikama i opisima programa na adresu El Toro ul Konrada Babnika 24 61210 Ljubljana

• Prodajem nov ZX-Spectrum 48K sa svim priborom, kasetom i uputstvima te nov ULA-čip Zvati od 16—18h Tel 024/35-523

• Specijalni novogodišnji popusti Devpac 3M Beta bejzik Mega bejzik za svega 600 dinara Vučinić Vladimir Bul oktobarske revolucije 7—11 34000 Kragujevac

• SPEKTRUMOVCI!!! KOMPLET 10 Fourth protocol (3 programa) Hacker Marsport Red Arrows M A Bumper Bored of Rings (4 programa) Shadowfire. KOMPLET 11 Bounty Bob Strikes Back Match Fishing Beach Head 2 Super Brat (Match Point 2), Abu Simbel Neverending Story (3 programa) International Karate 1 2 Rale Freeman Cena jednog kompleta 600 din Radović Branislav, Sonje Marinković 14/4 21000 Novi Sad tel 021/25-682 ili 022/424-824 (vikendom)

• SPEKTRUMOVCI KOMPLET 12 Fairlight Cyclon Attack Cluedo J S Willy 3 Mountains of Ket Thunder Hawks Chambers of Horrors Grid Bomb Lords of Time Back to Skool Highrise Harry Am Football KOMPLET 13 Chaos Covenant G G Cricket Teloe 3D Vortex Soho Sex Quest S H Quiz Aquirmy Wormy Dragon Fire Arhon S Belte Dodge City Radović Branislav Sonje Marinković 14/4, 21000 Novi Sad tel 021/25-682 ili 022/424-824 (vikendom)

• Komodor 64 — hit programi sa kasetu HACKER S V S 3 ROAD RACE KNIGHT LORE TOUR THE FRANCE Specijalni popusti Moguća razmena Petković Dejan Milana Rakića 29, 11000 Beograd, tel 011/424-744

• Prodajem paket programa za Komodor 64 Cene po dogovoru Javiti se na adresu Mikloš Dimitrije Petra Drapšina 55 21000 Novi Sad tel 021/21-758

• Prodajem najnovije svetske hitova za Komodor 64 Cene programa od 50 din na više Na komplete dobivate popust i do 50% Sve informacije i narudžbe na tel 041/417-819

• COMMODORE 64 — Novi (i stari) hit programi za vaš Komodor Besplatan katalog nagradni programi brze i kvalitetna usluga. Kmet Flavio G Martinuzzi 11 51470 Umag

COMMODOREOVCI! I u Novoj godini — najnoviji i najjeftiniji programi u kompletima koji sadrže do 50 igara Komplet I Summer Games Pitstop 2 Hockey Komplet J Winter Games Summer Games 2 Exploding Fist Komplet K Flight Simulation 2 Speed Kong Activision Box Komplet L Rembo Zorro Kompleti staje 1000 do 1500 dinara plus kaseta Tražite katalog (50 d) sa naznakom za CBM 64 FUTURE ORION 41000 ZAGREB Rubelićeva 7 tel 041/417-052

• Prodajem Commodora VIC 20. Hitno Tel 034/215-912 Zvati svakim danom posle podne sem ponedeljka

• Komodorci prvi put na jednoj kasetu možete naći najbolje i najnovije disk i kasetne programe za Komodor 64 vrlo povoljno Izbor od preko dve hiljade programa brza i kvalitetna usluga Tel (posle 17h) 011/131-641 Adresa Andrišić Zdenko II bulevar 34/52 11070 Novi Beograd

• Novo! Po prvi put u Jugoslaviji trostruki razdelnik što znači da sa jednog kasetofona učitavate a čak na dva snimate To znači da u isto vreme pravite dva kopije i samim tim olakšavate rad oko presnimavanja Cena razdelnika sa poštarinom je 3200 din Vujović Đurica Ustanička 168 Beograd tel 488-5242

KOMODOR

• COMMODORE 64 — Neverovatan izbor kasetnih programa PIT-STOP 3 Kung Fu Master Skool Daze SVS 1 i 2 JSW 1 i 2 Pyjamerama 1 2 i 3 Elite Tir-na-nog Kokotoni Wilt Stellar 7 Sorcery Boxing Spitfire 40 Tour the France Rugby Eureka Flight sim 2 Exploding fist i mnogi drugi Uputstva za početnika Besplatan katalog Žuljević Kemal Braće Kosorić 13 72220 Zavidovići tel 072/674-441

• COMMODORE Profesionalni prevodi PRIRUČNIK (650) PROGRAMERS REFERENCE GUIDE (1300) GRAFIKA I ZVUK (800) DISK 1541 (700) KAKO DA PROGRAMIRATE C-64 (1000) 200 MAPA ZA SPRAJTOVE (850) MAŠINSKO PROGRAMIRANJE ZA POČETNIKE — prevod 1985 g (1300) Profesionalni prevodi uputstava za uslužna programe SIMONS BASIC (700) VIZAWRITE (800) EASY SCRIPT (400) PRAKTIKALK (750) MAE (500). HELP 64+ (500) PASCAL (400) U kompletu (3000 d) Zbog velikog interesovanja za kompletnu literaturu cena svih 16 knjiga (10 000 din) Rok isporuke 24 časa. KOMPJUTER BIBLIOTEKA Filipa Filipovića 41 32000 Čačak tel 032/31-20

• COMMODORE 64 — Veliki izbor vrhunskih programa po super niskoj ceni od 30 din po programu Besplatan katalog razmena saradnja sigurna usluga Stanojević Nenad Filipa Filipovića 8 17500 Vranje

• Commodore 64
Kako u ovoj gomili oglasa naći onaj pravi??? Odgovor je kopija-softwara 50000 Dubrovnik Put bratstva i jedinstva b b
Preko 400 najnovijih i HIT programa Besplatan katalog

• Commodore — multisoft je obezbedio najnovije hitove za vašeg ljubimca Besplatan katalog isporuka u roku od 24h Dragan Jovanović Dragoslava Jovanovića 12 37000 Kruševac tel 037/30-568

• COMMODORE 64 — U Novu godinu sa novim hitovima Komplet programa JSV 2 CHIMERA BOXING LORDS OF MIDNIGHT REO MOON Za 100 din Tražite katalog Moguća razmena Dejan Rajić Sretena Mladenovića 7 11165 Beograd tel 530-637 ili 534-838

• Prodajem Commodore 64 kazetofon džojstik 800 programa resal literaturu Cena 110 000 Komodor 64 programi Najniža cena do 20din! Štalić Ronald Maksima Gorkog 8 42000 Varaždin

• Epson RX80 printer i interfaje EC-64 za Commodore 64 sve novo povoljno prodajem Dubravko Lisac Dž Bijedića 5B 47000 Karlovac tel 047/37-467

• COMMODORE 20 16 +4, 64 programi Tražite besplatan katalog! Šerman Šandor Rade Končara 23 23000 Zrenjanin

• Izrađujem različite hardverske dodatke za Commodore 64 CP/M kartica sa sistemskom diskatom i mnogo softvera 8 bitni A/D konverter univerzalni I/O port sa 4 slota za proširenja 80 kolonska kartica itd Tražite besplatan katalog Dražen Šćurić Klokočevac 105 43211 Predavac

• P N P elektonic
HARDVER ZA KOMODOR
NOVO IZ P N P electronica DODACI SADA I ZA KOMODOR 64 MEĐU PRVIM SIMONS BASIC NA KARTICI KOJA SE SPAJA NA USER PORT NEMA VIŠE DUGOTRAJNOG ČEKANJA DA SE UPIŠE SA TRAKE RADIMO JOŠ I LIGHT PEN A USKORO I I/O PORT TE CENTRONICS PRINTER INTERFEJS JAVITE SE SA VAŠIM ŽELJAMA BESPLATAN KATALOG NA ADRESU P N P electronic Jeretova 12 58000 Split

• Commodore 64 — Izbor od preko 2000 programa! Najnovija video igre uslužni programi novi programski jezici baze podataka muzika grafika obrada teksta i još mnogo toga a uskoro i još više! Sve i za svakoga i to na jednom mestu! Veliki izbor niske cene razmena popusti saveti za početnike! Tražite besplatni katalog na adresu Mirko Žagar Vukosavićeva 82 11090 Beograd

• Commodore 64 — postanite ekspert za paskal uz najnoviji interpretar paskala OXFORD PASCAL Program+kaseta +poštarina=850 d Capan Zdenački gaj 71 43293 Veliki Zdenci tel 046/67-025

• Najtraženiji programi za C-64 Jet Set Willy II Super Zaxxon II Flight Simulator II Staff of Karnath II, Boulder Dash II Hyper Circuit Elite Sorcery A View to Kill Sabre Wulf Jump Jet Gribblys Day Out Stop the Express Spitfire 40 + još tri programa + poklon hardverski dodatak Sve 10000 dinara Sekač Rajko Novi Šor 85 21466 Kucura

• COMMODORE 64 — NAJKVALITETNIJI NEOPHODNI I PROFESIONALNI PREVODI PROGRAMMER S REFERENCE GUIDE — 1300 d MAŠINSKI JEZIK ZA POČETNIKE — 1450 d GRAFIKA I ZVUK NA C-64 — 780 d UMJETNOST GRAFIKE NA C-64 — 900 d BASIC PRIRUČNIK — 660 d SIMON S BASIC — 660 d I PASCAL — 450 din Izvanredne kvalitete hitna isporuka na višestruke narudžbe popust 10%! DUŠKO BJELOTOMIĆ CENTAR 1 54550 VALPOVO tel 054/82-665 ili 041/883-141

COMMODORE 64
Najpopularnija igra januar 86
— Rambo 2
— School daze
— Commando
— Wizard s lair
— Who dare wins 2
— Superman
— Cylu
Programi + kasete = 1600 dinara
Despotović Milan Milana Zečara 5 11210 Beograd tel 011/712-442

• COMMODORE najnovija hitove prodajem i to po vrlo niskim cenama i uz programa prodajem i literaturu i rasel dirke POPUSTI NA PAKETE KATALOG BESPLATAN — – Roberto Krstović Radničko šetalište 44 58000 Split tel 058/46-832

NAJNOVIJI HIT PROGRAMI ZA COMMODORE 64
Ovo je prilika da nabavite samo najbolje programe po povoljnim cenama snimljene na uvoznim BASF kasetama.
KOMPLET 14
Break Fever Minolex Swag Seaworld Tir-na-nog Stop Express, Action Biker
KOMPLET 15
Blagger goes Hollywood Spitfire 40 Digital Drums Duango Bear Tread Tiger Shark Where My Bones
KOMPLET 16
Dummi Run Baseball II Dark Tever Arhipelago Black Knight Squah 3D Elidon
KOMPLET 17
Day After Flight Simulator II Tom Tur The France Sabre Wolf Speed King Demons of Topaz.
KOMPLET 18
Winter Games (sastoji se od 8 igara) Exploding Fist.
KOMPLET 19
Summer Games I (sastoji se od 8 igara)
KOMPLET 20.
A wiev To a Kill (4 igra sa Džems Bondom) Jet Set Willy II Barmu Bulders, Paradroid
KOMPLET 21
Summer Games II (8 sportskih igara)
KOMPLET 22
Far Hacker Skyjet Who Dares Wins International Tennis Cylu Mario s Brother Staff of Karnath I
KOMPLET 23
Beach Head II (4 ratne igre) Hyper Sports Scrolle Abadan Spy Hunter
KOMPLET 24
Super Hueye (4 igra najbolja simulacija sa helikopterom) Staff Of Karnath II Chimera, Hi Bouncher
Jedan komplet sa kasetom 1000 d Dva kompleta 1600 d Tri 2600 d Četiri 3400 d Pet 4200 d Šest 5000 d Svaki sledeći komplet još za po 200 dinara manje Spisak prethodnih 13 kompleta izašao je u oktobarskom broju Mog Mikra ISPORUKA U ROKU OD 24 ČASA Programe možete naručiti na adresu KRSTIĆ DRAGIŠA S J VUKOTIĆA 32 11080 BEOGRAD ili na telefon 011/533-611 (svakom ko naruči programe šaljem besplatan katalog sa preko 1500 programa)

• Prodajem i menjam programe za Komodor 64 Cena povoljne Milan Radovenović Bul Lenjine 165 11070 Novi Beograd

COMMODORE 64 Komplet! (sa kasetom 1600 – dinara)
komplet 1
who dares wins II
staff of karnath II
wizards lair
cylu (knight lore)
dynamite dan
chimera (alien 8)

komplet 2
skool daze
rambo II
superman
kung fu master
commando
willow patern
final synth samp

I još preko 2100 kasetnih i disketnih programa
Ćurčić Dušan Osmana Đikića 16a 11000 Beograd 011/762-022

• Komodorci! Najnoviji programi na kesati SORCERY HAKER TOUR THE FRANCE EXPLODING FIST uz najniže cene Tražite besplatan katalog Stanić Branko II bulevar 128 II ulaz 11070 Novi Beograd tel 122-545 ili 152-083

• Prodajem flopi disk 1541 i 1000 programa za Komodor 64 Gitić Saša Cerska 6 19320 Kladovo tel 019/87-569 ili 86-849

• Prodajem najnovija programe za Komodor 64 Isporuka odmah Tel 603-321 Maksimović Bana Goce Delčeva 34/28 11070 Novi Beograd

• Prodajem modul SIMON S BASIC za 15 000 dinara Tel 024/28-760

• Komodor SUMMER GAMES 2 EXPLODING FIST PITSTOP 2 DONALD DUCK SPITFIRE 40 ELITE THEATRE DUROVE STOP THE EXPRESS + 5 poklon programa za 1000 din Adrović Miroslav Milutina Uskokovića 43 24000 Subotica tel 024/29-760

• Jeftino programe za Commodora 64 prodajem za disk i kasetu MAGBET SORCERY SUMMER GAMES WINTER GAMES Kresoja Dragiša Vojvoda Stepe 418 11000 Beograd tel 475-419 radnim danom od 18—20 časova

• C-64 — Brian Bloodaxe Spy vs Spy II Elite Boulderdash II Pink Panter Everyone s Wally Jump jet Couldron Knockout II sa kazetom i poštarinom 1000 dinara Plaćanje pouzećem Miroslav Gekić Poljska 31 Strahoninac 42300 Čakovec

• Prodajam Commodora 64 + kazetofon +2 džojstika + 200 fantastičnih programa (SUMMER GAMES I I II EXPLODING FIST) Staro dva mjeseca Posenjak Dražen Gajeva 179 43405 Pitomača tel 046 782-019

BBC B

• BBC electron — multisoft Veliki izbor + mala cene = veliki razlog da naručite besplatan katalog Dragan Jovanović Dragoslava Jovanovića 12 37000 Kruševac tel 037/30 568

AMSTRAD

AMSTRAD
CPC 464 DO 6128

Nudimo vam veliki izbor od preko 200 najboljih fabričkih programa na kasetama i disketama po najpovoljnijim cenama Od poslovnih programa svih vrsta i nemana izdvajamo Word Star Pascal Fortran Micro Script Master File Mini Office Pružamo vam izvanrednu priliku da nabavite najnovije igre za vaš računar F B Boxing Beach Head Street Hawks Everyone s a Wally Exploding Fist Chiller Locomotion Android One, Dark Star te još mnogo novine za vaše kućne ljubimce Uverite se u našu profesionalnost i odlične usluge Za katalog i detaljna informacija obratite se na tel 021/20-742 Veselinov Đorđa M Gorkog 8 21000 Novi Sad

• Amstrad — multisoft
Najnoviji najbolji programi po najnižim cenama A VIEW to a Kill Super Pipeline 2 Everyone a Wally Trypods Rocky Horror Show The Way of Exploding Fist Crystal Theft Dragan Jovanović Dragoslava Jovanoviča 12 37000 Kruševac tel 037/30-568

• AMSTRADOVCI! Svi programi igre uslužni copy profesionalno prenimljeni direktno iz kompjutera Samo po 50 dinara Najjeftinije u Jugoslaviji! Tražite besplatan katalog Trajkovski Pende Kruma Spirkoski 7 97500 Prilep tel 098/28238

• Prodajem SCHNEIDER CPC 464 sa zelenim monitorom i sa preko 120 programa za 15 starih miliona Jovanović Dragan Dragoslava Jovanoviča 12 37000 Kruševac tel 037/30-568

AMSTRAD CPC 464 (SCHNEIDER) — NAJKVALITETNIJI ODABRANI PROFESIONALNI PREVODI UPUTSTVO ZA RAD NA AMSTRADU — 1100 d LOCOMOTIVE BASIC — 1200 d MAŠINSKO PROGRAMIRANJE ZA POČETNIKE — 1300 d UPUSTVA ZA DEVPAC I TASWORD PO 550 d GRAFIKA I ZVUK NA CPC 464 — 850 din Izvanredna kvaliteta hitna isporuka ne višestruke narudžbe popust 10%! DUŠKO BJELOTOMIĆ CENTAR 1 54550 VALPOVO tel 054/82-885 ili 041/663-141

• Bogata ponuda programa za Amstrad CPC 464 Jedan program 90 d Saša Volerič, Jušići 127, 51213 Jurdani

• AMSTRAD Profesionalni prevodi PRIRUČNIK CPC 464 (1100) LOCOMOTIV BASIC (1200), MAŠINSKO PROGRAMIRANJE (1300) ZAJEDNO (3400) Prevedena uputstva za uslužne programe DEVPAC TASWORD PASCAL QUILL MASTERFILE HOME BUDGET Pojedinačno (700), KOMPLET (3500) Snimljeni programi (200) AMSTRAD FUTURE Bate Jankovića 79 32000 Čačak tel 032/30-34

• OMEGA SOFT vam i dalje nudi veliki broj programa za Amstrad pojedinačno i u kompletima Specijalno za ovaj mesec Winter Sports osam različitih programa sa skijanjem hokejom klizanjem bobom i dr Adresa Đurić Slaviša Vladimira Gaćinovića 18 11000 Beograd

• Prodajem ŠNAJDER CPC 6128 sa zelenim monitorom (CP/M 2, CPM+ GSX, DR LOGO), ocarinjeno Tel 024/30-341

• HARRIER SOFT vam nudi programe za Amstrad po najnižim cijenama s najpovoljnijim uslovima Komplet programa za presnimavanja (TAPE TO DISK DISK TO TAPE DISK TO DISK I TAPE TO TAPE) Cijena komplete + poštarine + kazete iznosi 1500 din Naručite besplatan katalog cijelih kompleta i programa Adresa Leo Lugović Viktora Kovačića 26 41000 Zagreb, tel 041/676-683

• AMSTRADOVCI BINGSOFT predstavlja PLATINENKIT Izrada štampanih pločica Voice Chess Donald Duck Jammin Hockey Nibbler Super protoni, Millioner Wild Bunch Tripods Chomper Artist 50 din katalog Trumbićeva 14 41000 Zagreb tel 670-678

QL

• QL SOFT ima skoro sve postojeće programe za Sinclair QL Pune garancije za programe Tranutna isporuka Besplatan katalog Novitet Super bejzik kompajler Uskoro Fortran 77 kompajler Petković Dejan Dušana Dugolića 6 Beograd tel 404-690 ili Boško Gavović tel 633-501

QL
Prijevodi uputa
QL ARCHIVE (1500 d) — inteligentna baza podataka
QL TOOLKIT (1200 d) — proširenje SuperBASIC-a
Koristne programe za QL jezike asemblera monitora nove verzije programa šah itd i literaturu SuperBASIC mašinsko programiranje MC68000 tehnička priručnika itd razmjenjujem ili prodajem (program s mikrokazetom 2400 d)
Leon Kune Mihenovičeva 18/3, 43500 Daruvar tel (043) 31-883

• QL PROFESIONALNI PREVODI QL ARCHIVE (1500) QL-TOOLKIT (1200) QL-PASCAL (1500) QL-FORTH (1500) U KOMPLETU (5000) Zarić Slobodan, Bate Jankovića 78 32000 Čačak tel 032/30-34

• QL SOFTWARE — razmena i prodaja najnovijih programa! Najjeftinije cijene i veliki izbor Katalog! Tel 068/44-772 Tomaž Gorec Šentlovrenc 21 68212 Veliki Loka

RAZNO

• Prodajem Sensorstick — Senzorski džojstik osam pravaca automatska vatra PAT-JOY — senzorski džojstik sa daljinskim upravljanjem Nevenko Zrno posta mataniće 78000 Banjaluka

• „Electron news" opet sa vama! Sheme predajnici light show lečenje magnetskim poljem elektronska akupunktura pojačala vu-metri ispravljači alarmi Katalog 20 din Electron news 54000 Osijek, Kalnička 28A

• Hitno prodajem palicu za igru sa džekom za Komodor Atari ili Amstrad Četiri pravca pravo četiri ukoso i pucanje sa vrha palice Cena 3 200 din pouzećem Milošević Ivan Kosovska 107, 37000 Kruševac tel 037/24-658

• KAKTUS JOYSTICK — Novo na našem tržištu Originalan dizajn i prilagodljivost ruci Tražite podrobnije informacije HARD SOFT, P P 163 51000 Rijeka

• Prodajem nov ocarinjen Atari 800 XL sa kasetofonom palicom Dulckshot II i 30 programa Romić Zoran Aleksandra Tajkova 13/2 56300 Slavonska Požega tel 066/75-837 od 8 do 13 časova

• ŠPICA HARDWARE
Spektrum Profesionalna tastatura
ŠPICA GeneMons EPROM kartridž
EPROM programer
Komodor 64 CP/M modul sa uslužnim CP/M programima PS 232 ŠKD Forum Mikroradionica ŠPICA Kersnikova 4, 61000 Ljubljana

Literatura

COMET SOFTWARE

NAJVEĆI IZBOR LITERATURE KOD NAS

Preko 70 knjiga iz svih oblasti računarske tehnike
— o programskim jezicima
— programiranju
— operativnim sistemima
— mikroprocesorima
— hardveru itd
Velik izbor knjiga u originalu
Sve što želite da pročitate o vašim ljubimcima nađićete kod nas
— SPEKTRUM
— QL
— BBC
COMMODORE
— APPLE II

MILOVANOVIĆ LJUBIŠA
Petra Lekovića 57 11030
BEOGRAD
tel 011/558-007 posle 17 časova

RAZMENA

Ženska osoba prijatne spoljašnjosti 84K/100/320 lečaka Traži poznanstvo odgovarajućeg kompatibilnog partnera dobro hardverski opremljenog radi skidanja zaštite sa njenih programa i zajedničkih prijatnih softverskih tranutaka U obzir dolaze samo ozbiljne ponude Romantične sam prirode i volim sve što vole mladi Javi se čeka nas zajednička sreća

šifra „VIRGINSOFT"

ATARI ST520

SOFTWARE Poslovni programi
– Jezici
– Igre
LITERATURA
HARDWARE Centronics paralelni interfejs za povezivanje računara sa štampačem (Epson, Star Panasonic itd.)
– RS232 kabel
– Proširenje memorije na 1 Mbyt (2 Mbyt u razvoju)
SERVIS Servis manjih opravki
IZRADA PROGRAMSKE OPREME za organizacije udruženog rada i ustanove
Informacije HARDWARE SERVIS, Verje 31A, 61215
M[illegible] tel (061) [illegible] 548 u sredu i nedelju

Jedan glas za ZX81

Vašu reviju pratim od prvog broja i veoma mi se svidela Najviše mi se sviđa rubrika „Periferna oprema" u kojoj opisujete razne dodatke Čini mi se da praviše stranica posvećujete velikim kompjuterima koji su dostupni retko kojem Jugoslovenu Voleo bih kada biste objavljivali više tekstova o ZX81 koga sam nabavio nedavno ali o njemu ima vrlo malo tekstova, kao i praktičnih uputstava za rad sa njim Takođe bih želio da se u jednom od narednih brojeva pojavi umetak „sva rutina za ZX81" isto tako mi se čini da bi trebalo da objavljujete veći broj malih oglasa Ako znate adresu nekog ko prodaje pravod uputstva za ZX81 molim vas objavite ja

Drugarski vas pozdravljam

Dimitrije Šijan
Brigadira Ristića 46
23272 Novi Bečej

Dimitrije, sve što možemo da uradimo za stari, dobri ZX81 jeste da objavimo tvoje pismo. Nadamo se da će ti ovaj besplatni „mali oglas" pomoći da ti se jave istomišljenici i ljudi koji se bave računarom ZX81.

Kako s novim ROMom?

Interesuje me GENS koji se može nabaviti kod vas (u epromu) i novi ROM za spektrum U vašem časopisu sam pročitao da je novi ROM kompatibilan sa svim programima koji ne koriste interfejs 1 Molim vas da mi to malo objasnite Sam GENS upotrebljava interfejs 1 jer radi sa mikrodrajvom pa mi je sve to malo čudno Može li se sa novim ROM-om raditi sa mikrodrajvom i sa RS 232 a ako ne može, čemu onda novi ROM?

Uršič Ignac
Stari trg 247
62 380 Slovenj Gradec
(Slovenija)

Priča je prilično dugačka Ukratko kompletan projekat namenjen je onim spektrumovcima koje ne interesuju interfejs 1 i mikrodrajv



VRUĆE, HLADNO, VRUĆE

Poštovana redakcijo

Povodom napisa druga Dejana Ristanovića Dan po jutru, a jutro tmurno" u broju 10 Računara i recenzije programa VRUĆE LETOVANJE", molimo Vas da u sledećem broju časopisa Računari objavite sledeće

Smatramo da je osnovni red u prikazu jednog programa navesti sva one koji su učestvovali u izradi tog programa i čija imena stoje u programu i na uputstvima uz program

Autor grafike i dizajna programa „VRUĆE LETOVANJE" je Igor Kordej evropski poznat crtač stripova što je u pomanutom prikazu ispušteno

Zahvaljujemo na pogrešci kojom nas drug Ristanović imenuje dobitnicima prve nagrade na prvom Galaksijinom konkursu za najboljeg programera No treba reći da smo na tom konkursu dobili drugu nagradu

Pitamo se da li je to bilo tako davno da su uspomene druga Ristanovića i redakcije časopisa Računari toliko izbledele da se ne sećaju redosleda koji su sami odredili

Tvrdenje da su igre avanture jedan od glupljih načina za utucavanje slobodnog vremena je odličan primer totalne odsutnosti objektivnosti Samo nekoliko paragrafa iznad ove tvrdnje drug Ristanović izjavljuje da ima rejting deadly u igri Elite Pošto je to igra avanture (mada ne tekstualne jer pripada tipu „arcade adventure) čitaoci mogu sami izvući zaključak na koji način drug Ristanović utucava svoje slobodno vreme

Ili možda drug Ristanović tvrdi da su igre arkadnog tipa jedan od pametnijih načina za utucavanje slobodnog vremena ?

Uvek sam bio protiv ovakvih avantura jer slike nemaju nikakav funkcionalan značaj iznosi ispravan stav druga Ristanovića Slike su tu da smanje jaz između računara i korisnika-igrača i da daju nov kvalitet programu

Ovo je jedan od prvih odmačih programa ovog tipa Setite se koliko je avantura u Engleskoj napravljeno dok na red nisu došle ona kod kojih slike zavise od situacije Da je ovo strana domaća avantura podržali bismo mišljenje druga Ristanovića ovako smatramo da je deplasirano

„Da biste Srećku, junaku VRUĆEG LETOVANJA naložili da uzme nešto morate biti do kraja precizni UZMI GVOZDENI KLJUČ se ne može zameniti ni sa UZMI, ni sa UZMI KLJUČ pa čak ni sa UZMI GVOZDENI KLJUC (slovo 'č' dobijate kao 'SHIFT X' i na možete ga na nesreću zameniti sa c') je još jedna neobjektivna tvrdnja

Sve što je drug Ristanović ovom rečenicom rekao je istina Evo šta nam smeta u srpsko-hrvatskom jeziku se reč ključ piše sa 'č pa ne vidimo razloge da neko očekuje da se piše sa c Mi implementaciju slova specifičnih našem jeziku smatramo prednošću a drug Ristanović je upravo tome našao zamerku Bez veze!

Da bi neglasili neobjektivnost druga Ristanovića dovoljno je da kažemo da nije ni jednom rečju napomenuo da program igrača potpitanjima vodi ako unese nepotpunu rečenicu na primer

igrač OTVORI CR
program ŠTA?
igrač VRATA 'CR'
program KOJA?
igrač PREDNJA 'CR
program OTVORIĐ SAM PREDNJA

Ne znamo ni za jedan program koji se ovako inteligentno ponaša!

Činjenica je da VRUĆE LETOVANJE nije jednostavna igra i da bi se komandama koje istovremeno utiču na više objekata igrač vrlo brzo našao u situaciji koju ne može da kontroliše pa je stoga i obraćena pažnja da komande ne budu dvosmislene

Posebno je neprijatno što program ne prepoznaje naredbe koje su u krvi svih igrača avantura N S E W U D INV HELP moraju da se zamene sa IDI NAPRED IDI NAZAD ŠTA NOSIŠ (???) i tome slično" je još jedna besmislica neprimerena drugu Ristanoviću

Da možda drug Ristanović nije očekivao da odmači program namenjen domaćem tržištu prepoznaje engleske skraćenice? Ne vidimo šta korisniku može da smeta dosledna upotreba srpsko-hrvatskog jezika Mi to samo smatramo prednošću! Nismo toliki Englezi koliko se pravi drug Ristanović

Naredba HELP dobra prilika da autor programa izigrava Kirku čak nije ni implementirana je već traženje dlake u jajetu Tvrdimo da je mnogo važnije obezbediti druge kvalitete programa nego igraču u trenutku kada mu u određenoj situaciji ništa ne pada na pamet reći Baš ti dobro ide i

S druge strane izgleda da drug Ristanović mnogo voli da ga program iritira dok igra U tom slučaju preporučujemo mu da program IGRA

Spisak glagola koje računar poznaje je umereno siromašan, a broj poruke o greškama minimalan' ne može da prođe kao tvrdnja jer drug Ristanović izgleda nije ni pročitao da program razume sledeće glagole

ČITAJ, ČEKAJ, DAJ IDI, IZAĐI, JEDI, KUPI, OTVORI, OTKLJUČAJ, OSTAVI POPIJ, POLOMI PREGLEDAJ POPNI SE, PUSTI STAVI, SPAVAJ SIĐI TRČI, UZMI UĐI VOZI ZATVORI ZAKLJUČAJ, BROJ NOVAC GLEDAJ ŠTA NOSIŠ

Brojem to je 27 glagola Ne delimo mišljenje da je to „umereno siromašan broj

Posebno je iritirajuća tvrdnja da je broj poruka o greškama minimalan ' jer program u svakoj situaciji na svaku akciju igrača daje odgovarajuću poruku o grešci, ako je akcija igrača pogrešna Evo samo nekih od situacija do kojih može doći u igri

igrač IDI U SAMOPOSLUGU 'CR
program: SAMOPOSLUGA NE RADI
ili
igrač POLOMI ORMAN 'CR
program NE MOGU SUVIŠE JE TVRDO

„Isprobali smo raznorazne psovke i spectrum je odgovarao jedino NE RAZUMEM ŠTA ZNAČI

Cenimo vrlo bogat rečnik druga Ristanovića ali je podsticanje vulgarnog izražavanja zadnje što smo ovim programom želeli da izazovemo Da je program na ijednu psovku iz repertoara druga Ristanovića odgovorio na bilo koji način, to bi ga sigurno podstaklo da detaljno istraži ugrađeni rečnik psovki što bi tek bilo maltretiranje programa

S obzirom da nismo imali vremena da se iole ozbiljnije pozabavimo samom avanturom, teško možemo da sudimo o njenoj inventivnosti je izjava kojom se drug Ristanović ograđuje od rečenica koje slede u daljem tekstu

Smatramo da, budući da se nije iole ozbiljnije pozabavio samim programom, nije ni trebalo o njemu da piše Umesto toga, on nalazi gomilu zamerki izgleda da je drug Ristanović zaboravio da je VRUĆE LETOVANJE samo igra i to kako je dobro primetio, iz svakodnevnog, a ne realnog života

Ma koliko je Kontrabant siromašno urađen i neatraktivan, čini mi se da se u njemu krije mnogo više originalnih i duhovitih ideja Puno pravo na takvu tvrdnju ću jasno, imati tek kada dovedem Srećka na more tj nikada ' je rečenica od koje se čoveku jedino kosa može dići na glavi! Potpuno subjektivna izjava iza koje njen autor niti ne stoji što nam odmah i stavlja do znanja! (???)

Ne shvatamo kako drug Ristanović može ocenjivati program „VRUĆE LETOVANJE" kada je očigledno prema njegovoj sopstvenoj izjavi da nije video ni jednu dvadesetinu onoga što je u njemu sadržano!?

Mi bismo prema njegovom receptu mogli napisati

Čini nam se da je drug Ristanović pozeleneo od zavisti videvši naš program ali pravo na takvu tvrdnju imaćemo jasno, tek kada ga vidimo' kako ga igra tj. nikada"

Zbog svega nabrojanog molimo druga Ristanovića da razmisli o prikazu programa „VRUĆE LETOVANJE" u kome bi na objektivan način čitaocima časopisa Računari prikazao kako loše tako i dobre strane programa Molimo ga, ipak da pre toga igru IGRA!

Autori programa „VRUĆE LETOVANJE"

Ivan Gerenčir i Milan Pavićević

Ma šta neki ljudi rekli, uveren sam da rubrika Load „Dragi računari" nije uludo bačen prostor i da se iz nje može mnogo naučiti! Ja, na primer, već godinama živim u uverenju da je program bolji ako ima neku opciju ako se ta opcija nekome ne sviđa neka je ne koristi. Iz ovoga sam pisma, međutim, naučio da su dobre strane programa „Vruće letovanje ' što

*** Ne omogućava zamenu slova č, ć, ž i š slovima c, c, z i s koja se lakše kucaju**

*** Ne omogućava skraćivanje naredbi i izostavljanje njihovih delova**

*** Nema default vrednosti skoro ni za šta,**

*** Ne prepoznaje skraćenice N, S, E, W, U, D**

*** Nema naredbu HELP,**

*** Ne razume psovke**

Saznaje da je program koji nema razne stvari veoma vredan je poboljšalo i moje mišljenje o njegovim autorima Jer ni oni nemaju prvu nagradu na „Galaksijinom Konkursu.

Dejan Ristanović

alan tjuring

Računar se 1946. godine koristio jedino za balističke proračune, prognozu vremena i neka naučno-tehnička izračunavanja, a dvadeset godina kasnije primenjivao se u više od 600 oblasti ljudske delatnosti. Pre desetak godina govorilo se o 2500 računarskih profesija, a danas gotovo da nema posla čiji bi bar deo preuzeo računar. Da bismo mogli da govorimo šta nam valja sutra činiti, osvrnimo se na razvoj računara kroz priču o ličnostima koje su najviše doprinele njihovom razvoju i utrli put informatičkoj revoluciji.

Engleski inženjeri ne samo da su išli u korak sa svojim američkim kolegama, već su ih u mnogo čemu i preticali. Tako je CRT (cathode ray tube) — memorijski uređaj na bazi katodne cevi koji omogućava da se informacija pamti u obliku elektrostatičkog naboja na površini ekrana — izum naučnika sa mančesterskog univerziteta Viljamsa (F. C. Williams). U jesen 1947. Viljams je uspeo da obezbedi pamćenje 2048 (2K) bitova za nekoliko časova. Korišćenjem CRT mančesterski kompjuter „MARK I" uspešno je počeo sa radom 21. juna 1948. Osim toga, u mančesterskom računaru „MADAM" prvi put je bio primenjen INDEKS REGISTAR koji olakšava transformaciju naredbi u procesu rešavanja zadataka i koji je postao obavezan deo svih narednih računara.

Jedan od rukovodilaca projekta „MADAM" bio je poznati engleski matematičar Alen Tjuring koji je ranije aktivno učestvovao u izgradnji mašine ACE (Automatic Computing Engine — „engine" je ime dato u čast Bebidžu).

Tjuringovski metod

Alen Tjuring (Alan Turing) rođen je 1912. godine u Londonu. Sa šest godina samostalno je naučio da čita, piše i računa, sa jedanaest je vršio složene hemijske opite, a u petnaestoj godini sa olovkom u ruci proučavao je Ajnštajnovu teoriju relativiteta. „Ja mislim da će on biti matematičar" — pisao je njegov učitelj, „Takvi učenici kao što je on sreću se jednom u dvesta godina." Alan je zaista upisao matematiku na jednom od koledža Univerziteta u Kembridžu i ubrzo po diplomiranju 1937. godine publikovao je delo „O izračunljivim brojevima" („On computable Numbers") koje mu je donelo svetsku slavu.

Dr Robin Gandi se sećao: „U vreme prvih godina svog naučnog rada Tjuring se interesovao za niz pitanja, uključujući teoriju brojeva i kvantnu mehaniku, i počeo je da pravi mašinu za izračunavanje Rimanove zeta-funkcije. Interes za računanje doveo ga je na misao o svrsishodnosti razmatranja pitanja — kakve vrste procesa može da izvrši mašina i opisao je „univerzalnu" mašinu koja je, zahvaljujući odgovarajućim pravilima ponašanja, mogla da imitira ponašanje bilo koje druge mašine. Na taj način Tjuring je dao preciznu definiciju IZRAČUNLJIVOSTI i pokazao da postoje matematički zadaci čije rešenje NIJE IZRAČUNLJIVO u ovom smislu. Rad koji sadrži ova rezultate predstavlja tipičan tjuringovski metod, polazeći od osnovnih principa i koristeći konkretne primere on razvija teoriju koja ima karakter opštosti."

U toku poslediplomskih studija na Prinstonskom univerzitetu Tjuring se bavio matematičkom logikom. Po odbrani doktorske disertacije 1938. godine odbija da ostane u Americi kao asistent Džona fon Nojmana i vraća se u Englesku, gde postaje predavač na kraljevskom koledžu u Kembridžu.

HRONOLOGIJA

1912 Rođen u Londonu 23. juna
1937 Publikuje delo „O izračunljivim brojevima"
1939 Doktorira na Prinston univerzitetu i vraća se u Englesku. Postaje profesor Kembidžskog King koledža
1946 Počinje da se bavi pitanjima izgradnje praktičnih računskih mašina
1948 Dolazi na Mančesterski univerzitet i uključuje se u razradu prototipa „Mark I"
1950 U maju počinje da radi računar ACE koji je osmislio. Objavljuje Prvi Mančesterski programerski priručnik i znamenitu studiju koja izaziva burne diskusije — „Može li mašina da misli"
1951 Počinje primena prvog komercijalnog računara na svetu — Ferranti Mark I, čiji je programski sistem projektovao Tjuring
1952 Postaje jedan od rukovodilaca projekta računara „MADAM", u kojem je prvi put korišćen INDEKS REGISTAR
1954 18. juna pronađen je mrtav u svojoj postelji

Prvi programski sistem

Za vreme rata radio je u Britanskom departmanu veza vršeći niz računskih i teorijskih poslova za vojsku. Između ostalog, bio je u ekipi koja je uz pomoć elektromehaničkog računara „Colosuss" uspela da provali šifrovane poruke koje su Nemci slali pomoću svog remek-dela — Enigme. Počevši od 1946. godine Tjuring se bavi pitanjima izgradnje konkretnih računara, ali uporedo radi i na teorijskim naučnim istraživanjima. Njegov dolazak u Mančester stimulativno je delovao na programsku aktivnost. Tjuring je napisao prvi „Mančesterski programerski priručnik" (1951. godina), kada je objavio i knjigu „Računarska mašina i razum" („Computing Mashinery and Intelligence") preštampanu pod nazivom „Može li mašina da misli" („Can a machine think") u kojoj izvodi paralelu između rada mozga i dejstva računara. Ova studija, koja je izazvala burne diskusije među pionirima računarstva, ni danas nije izgubila na aktuelnosti.

Tjuring je takođe tvorac prvog programskog sistema koji je implementiran u prvi komercijalni računar na svetu — Ferranti Mark I. Zahvaljujući programskom sistemu ovaj računar se mogao programirati na simboličkom (asemblerskom) jeziku umesto na mašinskom, kako su se jedino mogli programirati prethodni računari sa unutrašnjim programom. Sem toga, na Tjuringovu sugestiju u ovaj računar ugrađen je i generator slučajnih brojeva. (Ferranti Mark I je prvi komercijalni računar na svetu jer je bio na tržištu u februaru 1951. godine, dakle pet meseci pre UNIVAC-a kome se nezasluženo pripisuje prvenstvo.) Ko zna za koliko bi još izuma čovečanstvo danas bilo bogatije da Tjuring nije tragično završio život 1954. godine.

Čudak iz Londona

Alan Tjuring privatno ni izbliza nije tako poštovan kao Alen Tjuring — naučnik. Po rečima njegovih savremenika, bio je čudak koji je predstavljao neskladnu kombinaciju genijalnog dečaka i senilnog profesora. Bio je odbojan i zatvoren individualac koji ni najmanje pažnje nije posvećivao svom izgledu. Nemarno odeven, neočešljan i zapušten, stalno je izgledao kao da je tek ustao iz kreveta. U njegovom životu majka je bila jedina žena, a kako je ignorisao sve ljude koji mu nisu bili intelektualno ravni, imao je vrlo malo prijatelja. Kada je 8. juna 1954. godine nađen mrtav u svojoj postelji, njegova majka i malobrojni prijatelji nisu mogli da poveruju nalazima lekara koji su ustanovili samoubistvo trovanjem kalcijum-cijanamidom smatrajući da je po sredi nesretan slučaj. Međutim, činjenica o njegovom privatnom životu i neprilagodljivost opšteprihvaćenim normama ponašanja ukazuju da je to ipak bilo samoubistvo. Smrt je uspela da zaustavi i takav energičan i kreativan um kakav je imao Alan Tjuring, ali nije izbrisala njegove ideje i dela koja i danas osvetljavaju put svima koji kreću izazovnim stazama računarstva.

Navanka Spalavić